Mundo AMSTRAD

LOS MEJORES LISTADOS PARA TU MICRO

EDITA: 

·año·I·numero·I·

mundo precio PVP IVA INC 200 pesetas

ATROPELLO LIST

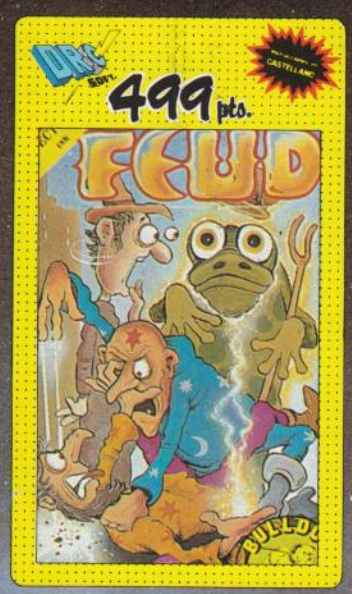
la amenaza del planeta SCORPIO list

SIMON list

los graficos definidos por el USUARIO

TRUCOS

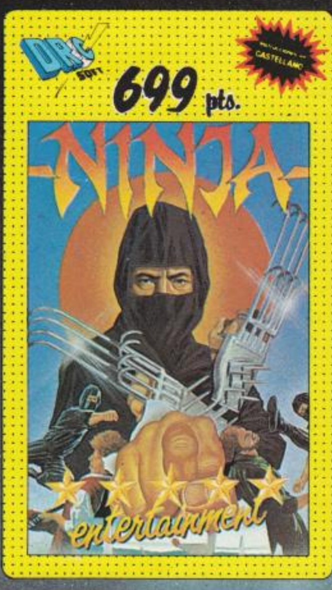
PVP CANARIAS 200 PTS



COMMODORE
SPECTRUM
AMSTRAD
MSX

FEUD
Con tu sabiduría puedes encontrar los ingredientes de las POCIONES MAGICAS, mientras recitas los conjuros de tu maligno enemigo LEANORIC.





COMMODORE
SPECTRUM
AMSTRAD

NINJA
Eres un NINJA, demuestra tu destreza luchando contra los mejores Budokas en un combate a MUERTE.

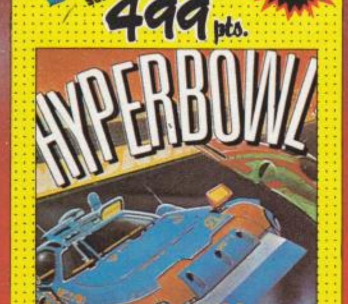


¡¡NO PUEDE SER!!

499 pts.

SERIE M.A.D. 699 pts.

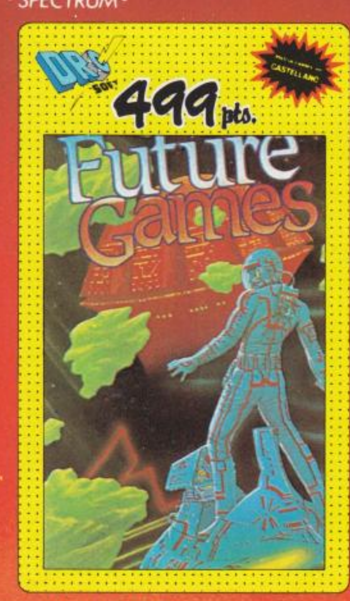
COMMODORE
SPECTRUM
AMSTRAD
MSX



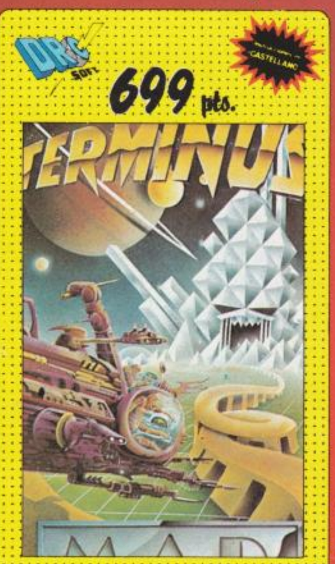


HYPERBOWL
En el año 3.600 el hockey sobre hielo ha alcanzado el nivel de máxima tecnología. Diez clases diferentes de naves, se enfrentan en un campo metálico...

· SPECTRUM ·



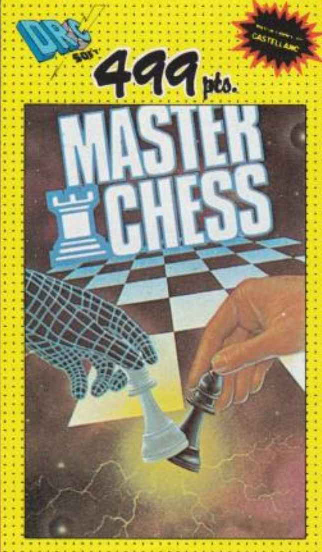
FUTURE GAMES
Juega tu libertad contra la muerte. Si ganas, la consigues, si pierdes, desde el lugar donde estés, ya no te importará...



TERMINUS
Un grupo de gamberros, que siembra el pánico por donde pasa, ha decidido rescatar a su lider del planeta prisión TERMINUS

SPECTRUM
AMSTRAD
MSX

COMMODORE



MASTER CHESS
Tu ordenador es tu mejor contrincante. Piensa más rápido que tu, pero tu puedes jugar mejor y ganarle. ¡INTENTALO!



Y otros 120 juegos mas.

MASTERTRONIC

EDITORIAL

Bienvenidos a este primer número de MUNDO AMSTRAD.

Esta nueva publicación va ha estar centrada mayoritariamente en los programas para microordenadores Amstrad.

Dedicaremos por tanto nuestra atención al mercado actual de los juegos comerciales, pero sobre todo, como bien podréis observar en páginas interiores, haremos especial hincapié en ofrecer un amplio repertorio de programas listados, que por su calidad os permitan aprender y/o perfeccionar vuestra técnica de programación de la manera más sencilla, útil y divertida posible.

Por supuesto, todos los programas que publicamos han sido probados por nuestros técnicos, por lo que garantizamos su buen funcionamiento. No obstante y dada la complejidad y extensión de algunos programas, os rogamos que tengais en cuenta las recomendaciones dadas en esta misma página.

Si aún así tuviérais algún problema, o simplemente quisiérais consultar o sugerirnos algo, no dudeis en llamar o escribir a nuestra redacción. Ahora bien, no podemos comprometernos a responder por correo personalizado a todas las peticiones, por lo que en los próximos números, abriremos una sección dedicada especialmente a este fin.

Asimismo también podréis contar en números sucesivos con un espacio gratuito reservado a la compra, venta y cambio directo entre lectores así que desde ya esperamos vuestros anuncios.



4	TRUCOS	
5	SOFTWARE	Los gráficos definidos por el usuario
7	UTILIDADES	Creador de laberintos
9	LIST	La amenaza del Planeta Scorpio
17	JUEGOS	Mermaid Madness. Nonamed Don Quijote. Viernes 13
23	LIST	Simón
26	LIST	Atropello
31	LIST	Mosquitos
18	LIBRERIA	

RECOMENDACIONES PARA TECLEAR UN PROGRAMA

Para aquellos que no estéis muy acostumbrados a teclear programas, os daremos a continuación una serie de instrucciones o pasos a seguir para que os resulte más sencillo y las probabilidades de que se produzcan errores en la ejecución de los programas sean menores:

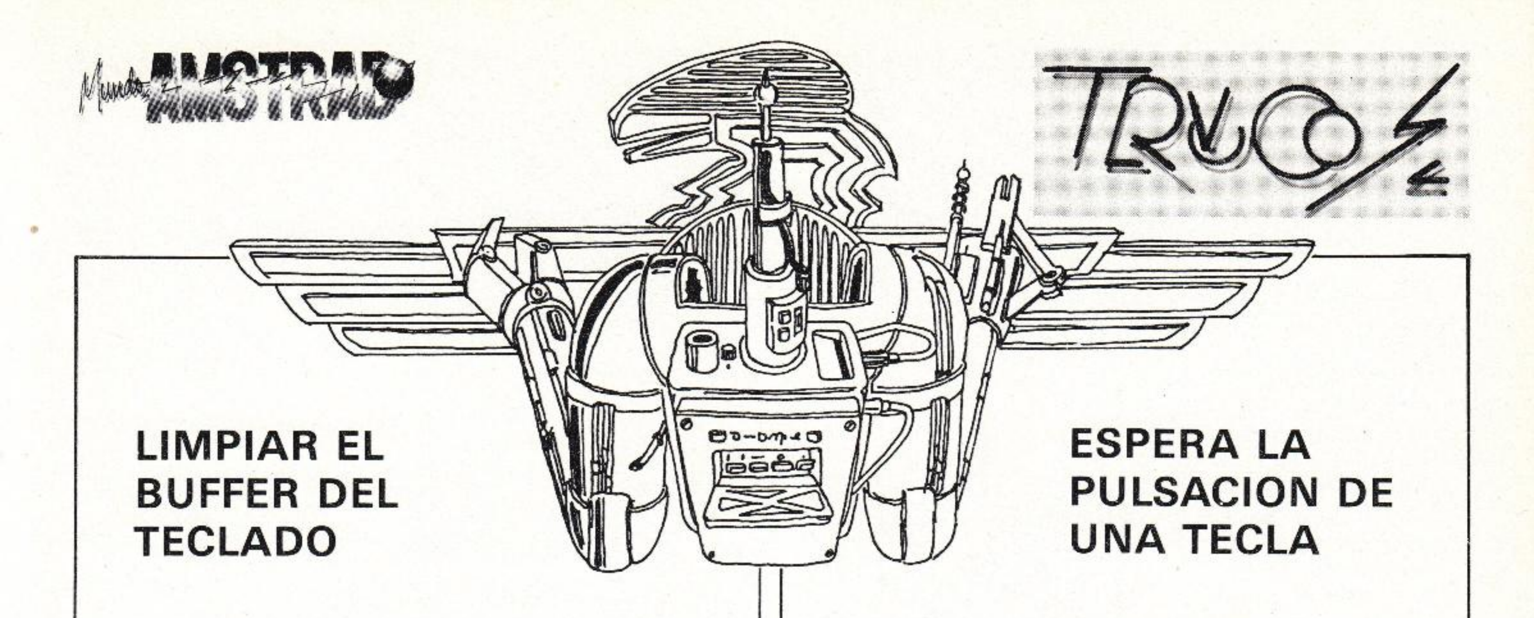
1. Como muchos ya sabréis, las líneas que después de su número correspondiente tienen una instrucción REM o un apóstrofe, no son consideradas por el programa sino que únicamente sirven como comentario aclaratorio para el programador y por lo tanto no es necesario que las copieis completas cuando aparezcan en nuestros listados, en cambio si es conveniente que pongáis el número de línea y la instrucción REM o ' (ejemplo 200 REM o 200') para evitar que puedan producirse errores al ejecutarse en el programa alguna instrucción de salto (GOSUB, GOTO, etc.)

2. Es fundamental que los números de línea los copiéis exactamente como figuran en el programa original.

3. Es conveniente, sobre todo en programas largos, salvar periódicamente a cinta o disco la parte que llevemos tecleada del listado antes de probar su funcionamiento para evitar la pérdida completa del programa en caso de que exista un error en el código máquina o bien se produzca un corte accidental de corriente, etc.

Dichas grabaciones periódicas es aconsejable no efectuarlas sobre la copia anterior o con el mismo nombre en caso de hacerlas en disco.

Mundo Amstrad es una publicación del Grupo Editorial SYGRAN S.A. Polig. Ind. Valdonaire. C/ Apolonia Hernández. HUMANES (Madrid). **Director:** Angel Herrero Fernández. **Director Adjunto:** Andrés Franco. **Director Técnico:** Luis Sanguino. **Coordinador Software:** Angel García. **Secretaria Redacción:** Mercedes Matons. **Colaboradores:** Antonio García, Angeles Guindal, Carmen Torres, J. Vicente Pons. **Diseño e Ilustraciones:** Javier Caballero. **Publicidad:** GENESIS, Tomás López, 3-6°, 28009 Madrid, Tel. 401 77 54 **Fotocomposición:** Fermar C/. Sílfide, 10. **Imprime:** Graficas Osiris S.A. Brañuelas, 29. **Distribuye:** G.M.E., plaza Castilla, 3. Madrid. Depósito Legal: M- 31873 - 1987.



LIMPIAR EL BUFFER DEL TECLADO

Limpiar el buffer del teclado. ¿En cuantas ocasiones te ha sido necesario limpiar el buffer del teclado antes de usar la función INKEY$? Seguro que muchas, por lo que a continuación te indicamos la manera de hacerlo:

```
10 ' *************************
20 ' * LIMPIA BUFFER TECLADO *
30 ' *************************
40 '
50 MODE 1
60 PRINT "PULSE TECLAS HASTA OIR
EL PITIDO"
70 FOR A=1 TO 5000
80 NEXT
90 PRINT CHR$(7)
100 PRINT "BASTA..."
110 PRINT
120 FOR A=1 TO 1000
130 NEXT
140 END
```

—Ejecútalo con RUN y pulsa varias teclas hasta oir el pitido.

—Puedes comprobar como el buffer se ha llenado con las teclas que has pulsado.

—Ahora introduce la siguiente línea en el programa.

```
135 CALL &BB03
```

—Vuelve a ejecutarlo con RUN y pulsa teclas de nuevo.

—Comprobarás que ahora el buffer queda vacío, puesto que en la línea 135 lo que hacemos es una llamada a la rutina KM RESET del sistema operativo situada en la dirección & BB03, que no necesita condiciones de entrada, y que reinicializa en las indirecciones director de teclado y buffer.

ESPERA LA PULSACION DE UNA TECLA

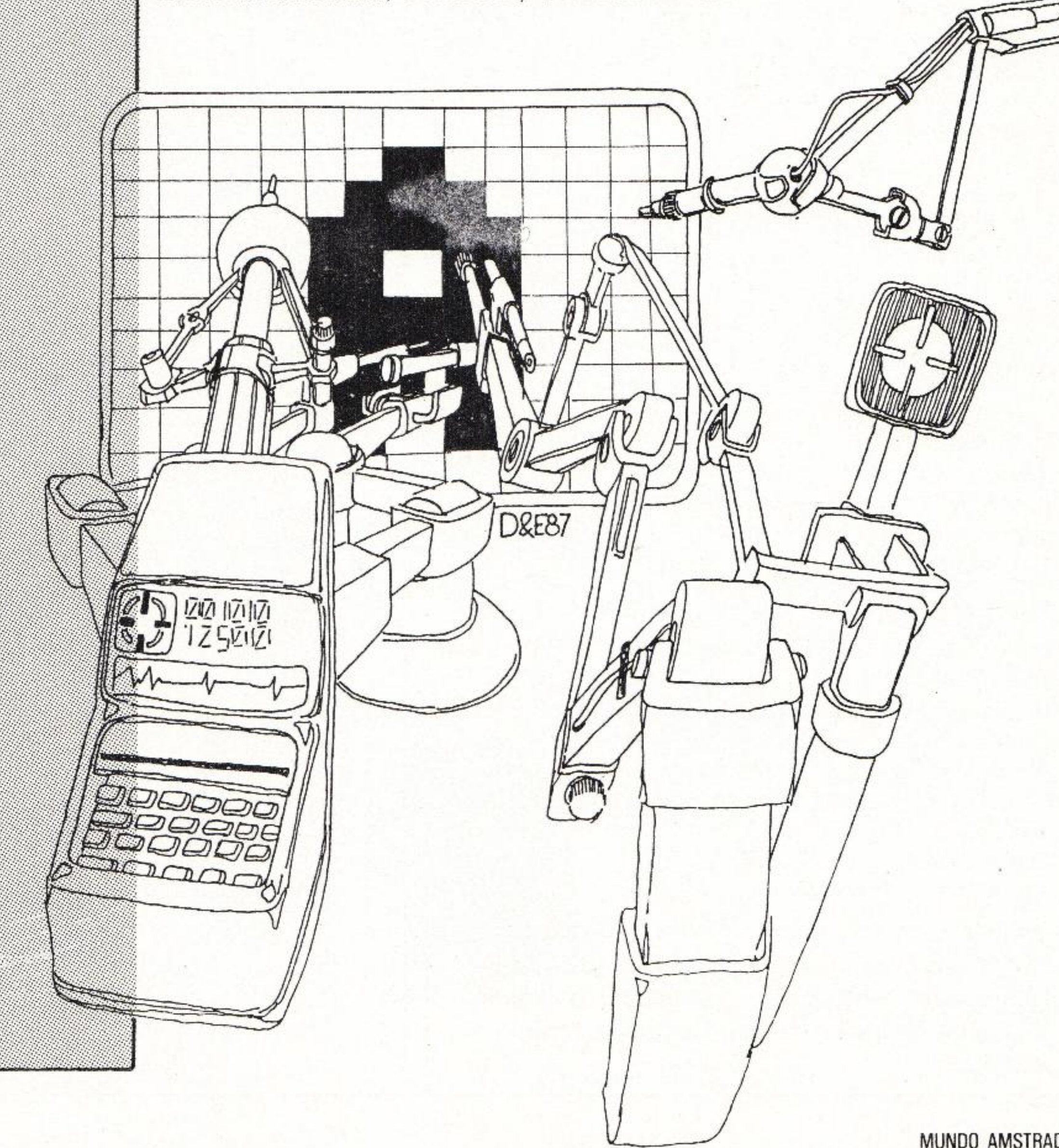
Espera la pulsación de una tecla. Seguramente conocerás varias maneras de hacer esto, mediante el empleo de la función INKEY$, pero la que nosotros proponemos tiene la ventaja de ser más sencilla, y de ocupar menos memoria. Un sencillo CALL &BB18 detiene el programa hasta que una tecla sea pulsada. Copia el ejemplo para verlo en funcionamiento:

```
10 '*****************************
20 '* ESPERA A PULSAR UNA TECLA *
30 '*****************************
40 '
50 MODE 1
60 PRINT RND
70 PRINT
80 PRINT "PULSE UNA TECLA PARA
SEGUIR"
90 PRINT
100 PRINT
110 CALL &BB18
120 GOTO 60
```

Lo que hacemos mediante el CALL &BB1 es llamar a la subrutina del Sistema Operativo, KM WAIT KEY, que no necesita condiciones de entrada, y que espera un código del buffer del teclado.

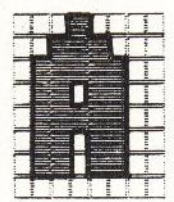
LOS GRAFICOS DEFINIDOS POR EL USUARIO (I)

El Amstrad dispone de un banco gráfico que consta de 256 caracteres (0-255), de los cuales, a partir del 32, son redefinibles, esto es, el usuario puede modificar los datos que definen la forma de cada carácter, para lo cual se disponen de dos instrucciones, SYMBOL y SYMBOL AFTER.



Antes de continuar con la explicación de estas instrucciones, dejemos claro qué es un carácter. Un carácter es una definición gráfica de 8 puntos de alto por 8 puntos de ancho, (ver figura 1).

Fig. 1



Para almacenar la forma de cada carácter, el ordenador emplea 8 bytes, 1 para definir cada una de las ocho filas. Este byte es un nº que en notación binaria indica, si el dígito es 1, que el punto va a ser coloreado de tinta, y si es 0, que va a llevar el color del papel.

La letra de la figura 1, se define con los siguientes datos:

		DECIMAL		BINARIO
FILA 1	--->	16	--->	00010000
FILA 2	--->	56	--->	00111000
FILA 3	--->	124	--->	01111100
FILA 4	--->	108	--->	01101100
FILA 5	--->	124	--->	01111100
FILA 6	--->	108	--->	01101100
FILA 7	--->	108	--->	01101100
FILA 8	--->	0	--->	00000000

Los datos de los 214 caracteres definibles (u.D.G.s), son almacenados, según el código ASCII de cada carácter. Por lo tanto, la tabla que contiene los gráficos definibles, tendrá una longitud de 214 × 8 = 1712 bytes.

¿Pero en qué lugar de la memoria es almacenada esta tabla?

Para conocer la dirección de comienzo de esta tabla, hemos de hacer uso de la función HIMEM, que nos devuelve la posición más alta de memoria utilizable para el BASIC.

La forma de empleo es la siguiente:

TABLA = HIMEM + 1

... y la variable TABLA contendrá el valor que buscábamos.

Este valor no es fijo, y varía cuando usamos la ilustración MEMORY, que fija el tope para el BASIC.

Para poder cambiar MEMORY es necesario eliminar todos los gráficos definidos, con la instrucción:

SYMBOL AFTER 256

de lo contrario, se producirá el error IMPROPER ARGUMENT. Por consiguiente, la instrucción SYMBOL AFTER n, donde n es un nº de 32 a 256, sirve para indicar, a partir de qué gráfico, se van a poder redefinir. Ejemplos:

SYMBOL AFTER 65	**Nos permitirá definir todos los gráficos a partir de la A.**
SYMBOL AFTER 32	**Nos permite redefinir todos los gráficos.**
SYMBOL AFTER 256	**Elimina los gráficos redefinidos, permitiéndonos así cambiar con MEMORY el tope de la memoria utilizable para el BASIC.**

Una vez reservado el espacio en memoria para definir los gráficos, podemos ya introducir las nuevas definiciones con la instrucción SYMBOL n, lista de 8 datos. Por ejemplo, para cambiar el carácter A por la bola de la figura 2, tendríamos que hacer lo siguiente:

SYMBOL AFTER 65 Para hacer el carácter A redefinible.
SYMBOL 65, 60, 126, 255, 255, 255, 255, 126, 60 Para dar la nueva definición al carácter A de código 65.

Ahora, cada vez que el ordenador tenga que imprimir una A. imprimirá nuestra bola.

Otra forma de hacer lo mismo sin emplear la instrucción SYMBOL, es pokear los datos en memoria. La forma en que tenemos que calcular la dirección donde comienza el carácter es la siguiente:

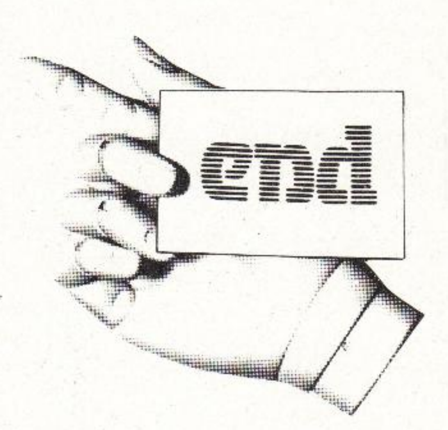
DIRECCION = HIMEM + 1 + 8 * (B - A)

...donde: A Es el código del carácter a partir del cual se ha hecho el symbol after (65 en el ejemplo).
B Código del carácter a definir.

En la variable DIRECCION, queda la dirección a partir de la cual hemos de almacenar los 8 datos que definen el nuevo gráfico.

En nuestro próximo número, incluiremos dentro de los listados, un programa con el que te será muy sencillo generarte grandes bancos de gráficos, y en esta sección una subrutina en lenguaje máquina, para conmutar bancos gráficos completos.

¡Hasta el próximo mes!



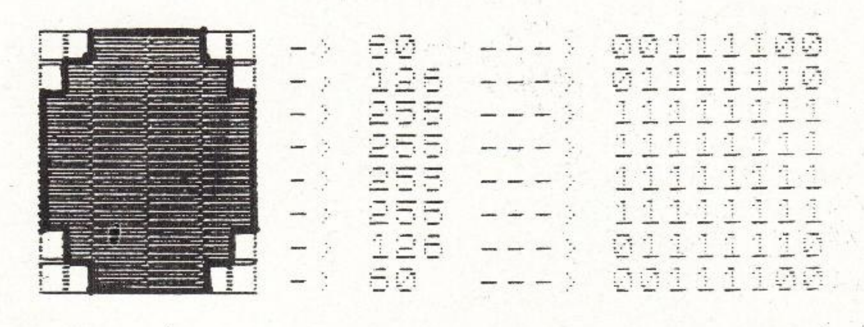


Figura.2

CREADOR DE LABERINTOS

Ofrecemos al aficionado programador de juegos, una herramienta, que le permitirá generar laberintos de una pantalla de tamaño (como máximo), con el uso de los cursores.

```
10 '
20 ' *** GENERADOR DE LABERINTOS ***
30 ' *** EMPLEA EL CARACTER "a"  ***
40 ' ***   PARA DIBUJARLOS       ***
50 '
60 ' *** SE ALMACENA A PARTIR DE ***
70 ' ***    LA LINEA 8000       ***
80 '
90 ' ***  MUNDO AMSTRAD  1987   ***
100 '
110 ' *** ANGEL GARCIA DELGADO  ***
120 '
130 ' * PRESENTACION *
140 '
150 SYMBOL AFTER 256
160 MODE 1
170 INK 0,0
180 INK 1,26
185 INK 2,11
190 PEN 1
200 BORDER 11
210 PRINT
220 PRINT" CREADOR DE LABERINTOS."
230 PRINT" ______________________"
240 PRINT
250 PRINT"     Este programa te permite g
enerar"
260 PRINT
270 PRINT" laberintos, empleando los cur
sores"
280 PRINT
290 PRINT" en cualquier modo de pantalla
, y"
300 PRINT
310 PRINT" a continuacion los transforma
a un"
320 PRINT
330 PRINT" programa, que prodras unir al
tuyo"
340 PRINT
350 PRINT" empleando el comando MERGE."
```

Para dibujarlos se empleará unicamente el carácter "a" redefinido, de forma que al añadirlo al juego, el carácter simulará ladrillos, piedras, etc...

Una vez esté dibujado el laberinto, pulsando la tecla "D" se mostrará en pantalla el programa necesario para dibujar en BASIC ese laberinto. Entonces subiendo el cursor, y pasando sobre todas las líneas pulsando COPY formará parte del programa actual. Para borrar el programa CREADOR DE LABERINTOS puedes hacer NEW antes de introducir las líneas que dibujarán el laberinto. O si ya las tienes introducidas emplea la instrucción DELETE 10-760.

CREADOR DE LABERINTOS

```
360 PRINT:PRINT
370 CALL &BB18
380 '
390 ' * REDEFINE LETRA "a" PARA DIBUJAR
EL LABERINTO *
400 '
410 SYMBOL [illegible] 97:SYMBOL 97,0,126,126,
126,126,126,126,0
420 INPUT " MODO DE PANTALLA: ",M
430 IF M<0 OR M>3 THEN 420
440 PRINT
450 INPUT " ANCHO:",AN
460 IF AN<1 OR AN>20+20*M THEN GOTO 450
470 PRINT
480 INPUT " ALTO:",AL
490 IF AL<1 OR AL>24 THEN GOTO 480
500 DIM A$(AL,AN)
510 MODE M
520 '
530 ' * PREPARA LA PANTALLA *
540 '
550 WINDOW 1,AN,1,AL:PAPER #1,2:CLS#1:CL
S
560 X=1:Y=1
570 IF NOT INKEY(0) THEN X=X-1:IF X<1 TH
EN X=1
580 IF NOT INKEY(2) THEN X=X+1:IF X>AL T
HEN X=AL
590 IF NOT INKEY(8) THEN Y=Y-1:IF Y<1 TH
EN Y=1
600 IF NOT INKEY(1) THEN Y=Y+1:IF Y>AN T
HEN Y=AN
610 LOCATE Y,X:PRINT CHR$(143);
620 IF NOT INKEY(47) THEN A$(X,Y)="a"
630 IF NOT INKEY(71) THEN a$(x,y)=" "
640 LOCATE Y,X:PRINT" ";
650 IF NOT INKEY(61) THEN 710
660 LOCATE Y,X:PRINT a$(x,y);
670 FOR G=1 TO 30:NEXT:GOTO 570
680 '
690 ' * CONSTRUCCION DEL PROGRAMA *
700 '
710 MODE 2
720 FOR A=1 TO AL:PRINT(7999+A);"?";CHR$
(34);:FOR B=1 TO AN
730 IF A$(A,B)="" THEN A$(A,B)=" "
740 PRINT A$(A,B);:NEXT:PRINT CHR$(34):N
EXT
750 CALL &BB00
760 END
```

Para que el programa generado ocupe otras líneas, no tendrás más que renumerarlo con RENUM.

Para unirlo a tu programa usa la instrucción MERGE, que sobrepondrá el último programa cargado al ya existente en memoria.

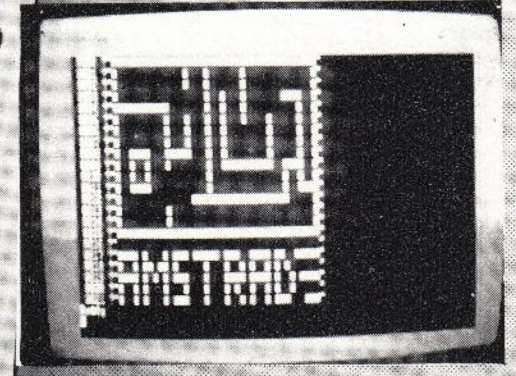
En lo referente al programa, la presentación ocupa las líneas 10-380.

A continuación se redefine la letra "a" con un gráfico que la distinga. Luego se te preguntará el modo de pantalla y las dimensiones del laberinto en caracteres.

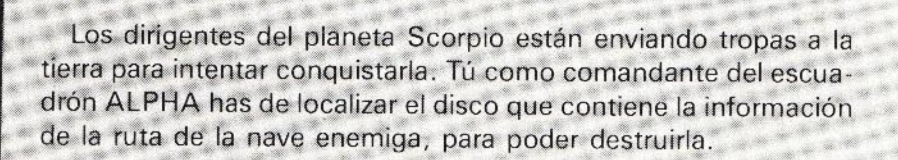
Seguidamente se dimensiona una tabla alfanumérica que contendrá el carácter de cada posición del laberinto. Se prepara la pantalla poniendo el modo requerido y definiendo una ventana del tamaño del laberinto.

Entramos luego en el bucle principal donde se leen las teclas del cursor para mover éste por la pantalla, el espacio para pintar, la tecla "z" para borrar y la "D" para obtener el programa.

La subrutina que genera las líneas de programa está localizada entre las 690 a 760.



LA AMENAZA DEL PLANETA ESCORPIO



Los dirigentes del planeta Scorpio están enviando tropas a la tierra para intentar conquistarla. Tú como comandante del escuadrón ALPHA has de localizar el disco que contiene la información de la ruta de la nave enemiga, para poder destruirla.

```
1000 REM *******************
1010 REM * LA AMENZA DEL    *
1020 REM * PLANETA SCORPIO *
1030 REM *******************
1040 REM
1050 GOSUB 2740
1060 MODE 1:INK 1,26:INK 2,2:INK 3,9:WIN
DOW 2,40,2,14:PAPER 3:CLS:WINDOW 1,39,1,
13:PAPER 2:CLS:GOSUB 2570:PEN 1:A$="L A"
:GOSUB 2550:A$="A M E N A Z A":GOSUB 255
0:A$="D E L":GOSUB 2550:A$="P L A N E T
A":GOSUB 2550:A$="S C O R P I O":GOSUB 2
550
1070 PAPER 0:WINDOW 1,40,16,25:GOSUB 257
0:A$="AUTOR: ANGEL GARCIA DELGADO":GOSUB
2550:A$="1987 MUNDO AMSTRAD":GOSUB 2550
:PEN 2:GOSUB 2570:A$="PULSE -S- PARA COM
ENZAR A JUGAR":GOSUB 2550
1080 IF INKEY(60)<>0 THEN SOUND 1,RND*10
0,1,10:GOTO 1080
1090 P=0:VI=0
1100 MODE 1:PEN 1:INK 1,26:LOCATE 1,5:A$
="FASE 1: DESPEGUE":GOSUB 2550:GOSUB 257
0:A$="CONTROLES:":GOSUB 2550:A$="CURSOR
IZQUIERDA":GOSUB 2550:A$="CURSOR DERECHA
":GOSUB 2550:A$="Q. ACELERAR":GOSUB 2550
:A$="A. FRENAR":GOSUB 2550
1110 GOSUB 2570:A$="PUNTUACION: 50 * NAV
E":GOSUB 2550
1120 GOSUB 3600
1130 N=6
1140 INK 2,6:INK 4,20:INK 0,0:PAPER 0:BO
RDER 0:MODE 0:INK 1,26:INK 3,9:PEN 3:LOC
ATE 3,25:PRINT"NA:";N;"       VI:";VI;
1150 PLOT -200,0,4:DRAW 200,200:DRAW 430
,200:DRAW 830,0
1160 PLOT 0,400:DRAW 200,300:DRAW 430,30
0:DRAW 630,400
1170 PLOT 200,200:DRAW 200,300:PLOT 430,
200:DRAW 430,300
```

Para salir a buscar el disco tendrás que sacar las naves de los hangares, y atravesar la tormenta de asteroides para llegar al planeta SCORPIO.

Para destruir la nave enemiga has de acertarle en el centro del ojo energético. Has de tener cuidado además, de no ser alcanzado por los lasers que te lanzaran las naves escolta.

Los controles se van indicando en cada fase de juego, asi como los puntos conseguidos tras su consecución.

Una vez conocida la temática, pasemos al comentario del programa.

— Líneas 1000-1080. Se inicializa el programa y se dibuja la portada de presentación. Con el GOSUB de la línea 1050 se salta a una subrutina que introduce los datos de una rutina de sonidos, y se redefinen los gráficos.

La presentación esta creada en MODO 1. Para dibujar los cuadrados, se definen ventanas del tamaño del recuadro a colorear, se fija el papel con el

LA AMENAZA

```
1180 LOCATE 9,13:PRINT"      "
1190 PLOT 254,200:DRAWR 0,70:DRAWR 124,0
:DRAWR 0,-70:DRAWR 10,0
1200 A=315:B=1
1210 PLOT 316,A-14,4:DRAWR -(A-200)-B*2,
0:DRAWR -B*2,-B:DRAW 316+200-A-B*4,220:P
LOT 316,A-14:DRAWR A-200+B*2,0:DRAWR B*2
,-B:DRAW 316+A-200+B*4,220:A=A+B:B=B+1:I
F A<450 THEN 1210
1220 PLOT 0,220:DRAWR 250,0:PLOT 380,220
:DRAWR 250,0
1230 PEN 2:FOR A=1 TO N-1:LOCATE 12+A,14
+A:PRINT"ab":NEXT A:PEN 1
1240 PO=1:X=450:Y=80:IX=0:IY=0:TAG
1250 XB=X:YB=Y
1260 IF NOT INKEY(1) THEN PO=PO+1:IF PO>
4 THEN PO=1
1270 IF NOT INKEY(8) THEN PO=PO-1:IF PO<
1 THEN PO=4
1280 V=(NOT INKEY(69))-(NOT INKEY(67))
1290 IF IX<>0 OR IY<>0 THEN SOUND 2,400,
25,10,0,0,31
1300 IX=IX-(V AND PO=1)-(V/2 AND PO=2)+(
V/2 AND PO=4)
1310 IY=IY+(V/2 AND PO=2)+(V AND PO=3)+(
V/2 AND PO=4)
1320 IF IX<-8 THEN IX=-8
1330 IF IX>8 THEN IX=8
1340 IF IY<-8 THEN IY=-8
1350 IF IY>8 THEN IY=8
1360 C=0:FOR A=0 TO 7:C=C+TEST(X+A*4,Y+2
)+TEST(X+(A+8)*4,Y+2):NEXT A
1370 IF C<>0 OR X<0 OR X>610 THEN X=X+16
:Y=Y-4:GOSUB 3300:GOTO 1440
1380 X=X+IX:Y=Y+IY:MOVE XB,YB:PRINT"  ";
1390 MOVE X,Y,1:PRINT CHR$(95+PO*2);CHR$
(96+PO*2);
1400 IF Y>255 THEN 1430
1410 XB=X:YB=Y
1420 GOTO 1260
1430 |ZAP:VI=VI+1:GOTO 1440
1440 N=N-1:IF N<1 THEN 1460
1450 GOTO 1140
1460 P=P+VI*50:GOSUB 3420
1470 REM
1480 REM **********
1490 REM * FASE 2 *
1500 REM **********
1510 REM
```

color deseado y se hace CLS de esa ventana. Los textos se cargan en la variable A$ y son centrados e impresos en la subrutina de la línea 2550.

Tras dibujar la presentación, se genera un sonido aleatorio en espera de la pulsación de la tecla S que da paso al juego.

—Línea 1090. Se inicializan los puntos a 0 (P = 0) y las vidas también (VI = 0).

—Líneas 1100-1120. Se imprimen las instrucciones de la FASE 1. Y en la línea 1120 se llama a una subrutina que toca una corta melodía antes de continuar.

—Líneas 1130-1230. Se inicializa el número de naves iguales a 6, se imprimen los marcadores y se dibuja el hangar con PLOT y DRAW.

En la línea 1230 se imprimen las naves que todavía quedan en el hangar.

—Líneas 1240-1460. Este bloque es el corazón de la FASE 1. Primero se inicializan las variables de la posición (PO), las coordenadas de situa-

```
1520 MODE 1:LOCATE 1,5:A$="FASE 2: ASTER
OIDES":GOSUB 2550:GOSUB 2570:A$="CONTROL
ES:":GOSUB 2550:A$="CURSOR IZQUIERDA":GO
SUB 2550:A$="CURSOR DERECHA":GOSUB 2550:
GOSUB 2570:A$="PUNTUACION: 500 PUNTOS":G
OSUB 2550:GOSUB 3600
1530 T=250:S=9:PO=2:MODE 0
1540 LOCATE 1,1:PRINT CHR$(8);CHR$(8)
1550 PEN 1:LOCATE S,25:PRINT CHR$(97+2*P
O);CHR$(98+2*PO);:SOUND 2,600,10,10,0,0,
31
1560 PLOT 630*RND,399,3:PEN 2:LOCATE 1+R
ND*19,1:PRINT"i";
1570 LOCATE S,24:CALL 30000:C=PEEK(29999
):LOCATE S+1,24:CALL 30000:A=PEEK(29999)
1580 IF C=105 OR A=105 THEN X=23:Y=S-0.5
:GOSUB 3280:GOSUB 3410
1590 PO=2:IF NOT INKEY(8) THEN PO=1:S=S-
1:IF S<1 THEN S=1
1600 IF NOT INKEY(1) THEN PO=3:S=S+1:IF
S>19 THEN S=19
1610 T=T+1:P=P+2:IF T>499 THEN |SQUELCH:
FOR A=1 TO 500:NEXT A:GOTO 1630
1620 GOTO 1540
1630 GOSUB 3420
1640 REM
1650 REM **********
1660 REM * FASE 3 *
1670 REM **********
1680 REM
1690 MODE 1:LOCATE 1,5:A$="FASE 3: INFIL
TRADO":GOSUB 2550:GOSUB 2570:A$="CONTROL
ES:":GOSUB 2550:A$="Q. ARRIBA":GOSUB 255
0:A$="A. ABAJO":GOSUB 2550:A$="CURSOR IZ
QUIERDA":GOSUB 2550:A$="CURSOR DERECHA":
GOSUB 2550:GOSUB 2570
1700 A$="PUNTUACION: 2500 PUNTOS":GOSUB
2550:GOSUB 3600
1710 MODE 1:I=0:GOSUB 2080:PEN 1:Y1=6:D1
=2:D2=1:X=19:Y=2:XB=X:YB=Y
1720 X=X+(INKEY(67)=0)-(INKEY(69)=0):Y=Y
+(INKEY(8)=0)-(INKEY(1)=0)
1730 LOCATE Y1,2:PRINT"  ":LOCATE Y1,3:P
RINT"  ":LOCATE 26-Y1,14+D2:PRINT"  ":LO
CATE Y1+1,17-D2:PRINT" ":Y1=Y1+D1:IF Y1>
22 OR Y1<3 THEN D1=-D1
1740 LOCATE Y1,2:PRINT"xy";:LOCATE Y1,3:
PRINT"z{";:D2=3-D2:LOCATE 26-Y1,14+D2:PR
INT CHR$(225):LOCATE Y1+1,17-D2:PRINT CH
R$(225)
1750 LOCATE Y,X:CALL 30000:C=PEEK(29999)
1760 IF C=108 THEN X=XB:Y=YB
```

ción (X e Y) y los incrementos para las coordenadas X e Y (5X, 5Y). Con TAG activamos el modo gráfico, para desplazar nuestra nave pixel a pixel.

De la 1260 a la 1420 se encuentra el bucle principal de la fase 1. Primero se leen las teclas que modifican nuestra dirección, y luego las que incrementan o disminuyen la velocidad.

Luego en caso de estar la nave activa se genera el sonido de los motores. A continuación se incrementan IX y 5Y para pasar a la nueva posición. Se comprueba si chocamos (línea 1360) con TEST y si no chocamos y no sobrepasamos los límites nuestra nave se imprime y repite el bucle.

En la línea 1430 se hace un sonido para decir que hemos sacado nuestra nave, y se incrementa el número de vidas. En la 1460 se suman los puntos, y se salta a la subrutina que nos informe de los puntos que llevamos y las vidas que nos quedan.

—Líneas 1470-1520. Se im-

```
1770 IF C=230 OR C=225 OR (C>119 AND C<1
24) THEN |EXPLODE:LOCATE YB,XB:PRINT"w":
FOR A=1 TO 500:NEXT A:GOSUB 3410:GOTO 17
10
1780 IF C=109 THEN I=1:|PING:GOTO 1870
1790 LOCATE YB,XB:PRINT" ":LOCATE Y,X:PR
INT CHR$(248+RND*3);
1800 XB=X:YB=Y
1810 GOTO 1720
1820 REM
1830 REM **********
1840 REM * FASE 4 *
1850 REM **********
1860 REM
1870 GOSUB 3420:MODE 1:LOCATE 1,5:A$="FA
SE 4: HUIDA":GOSUB 2550:GOSUB 2570:A$="C
ONTROLES:":GOSUB 2550:A$="Q. ARRIBA":GOS
UB 2550:A$="A. ABAJO":GOSUB 2550:A$="CUR
SOR IZQ.":GOSUB 2550:A$="CURSOR DER.":GO
SUB 2550:GOSUB 2570:A$="PUNTUACION: 10 *
 TIEMPO"
1880 GOSUB 2550:GOSUB 3600
1890 T=250
1900 MODE 1:I=1:Y=12:X=7:XB=X:YB=Y:GOSUB
 2080:PEN 1:LOCATE 1,23:A$="TIEMPO:     "
:GOSUB 2550
1910 X=X+(INKEY(67)=0)-(INKEY(69)=0):Y=Y
+(INKEY(8)=0)-(INKEY(1)=0)
1920 LOCATE Y,X:CALL 30000:C=PEEK(29999)
1930 IF C=108 THEN X=XB:Y=YB
1940 IF C=230 THEN |EXPLODE:LOCATE YB,XB
:PRINT"w":FOR A=1 TO 500:NEXT A:GOSUB 34
10:GOTO 1890
1950 IF C=247 THEN |ZAP:P=P+10*T:GOTO 20
50
1960 LOCATE YB,XB:PRINT" ":LOCATE Y,X:PR
INT CHR$(248+RND*3);
1970 XB=X:YB=Y
1980 LOCATE 22,23:PRINT T;" ":T=T-1:IF T
<0 THEN FOR A=1 TO 5:FOR B=1 TO 10:|PING
:NEXT B:FOR B=1 TO 250:NEXT B:NEXT A:|EX
PLODE:FOR A=1 TO 500:NEXT A:GOSUB 3410:G
OTO 1890
1990 GOTO 1910
2000 REM
2010 REM **********
2020 REM * FASE 5 *
2030 REM **********
2040 REM
```

primen las instrucciones de la fase 2.

—Línea 1530. Se inicializan las variables que se van a usar. El tiempo (= 250), nuestra posición (S = 9), y el gráfico de nuestra nave (PO).

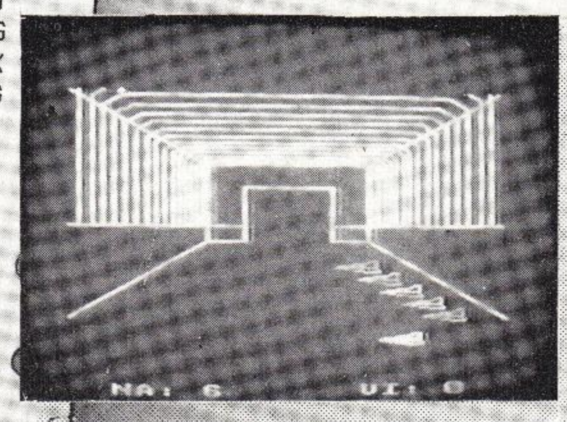
—Líneas 1540-1630. Lo primero que hacemos es provocar un scroll hacia abajo en la línea 1540. Inmediatamente después imprimimos nuestra nave y hacemos el sonido de sus reactores.

En la línea 1560 dibujamos la nueva línea de paisaje que aparece, una estrella, y un asteroide.

Luego comprobamos si nuestra nave colisiona con los asteroides. En caso afirmativo llama a la subrutina explosión y a la de restar 1 vida. En caso negativo se lee el teclado para saber si te mueves y a continuación se incrementa el tiempo que al ser 500 da paso a la fase 3.

—Líneas 1640-1700. Instrucciones de la fase 3 y la melodía.

—Línea 1710. Se dibuja el la-



```
2050 GOSUB 3420:MODE 1:LOCATE 1,5:A$="FA
SE 5: DUELO A MUERTE":GOSUB 2550:GOSUB 2
570:A$="CONTROLES:":GOSUB 2550:A$="CURSO
R IZQ.":GOSUB 2550:A$="CURSOR DER.":GOSU
B 2550:A$="SPACE. DISPARO":GOSUB 2550:GO
SUB 2570:A$="PUNTUACION: 50.000 PUNTOS"
2060 GOSUB 2550:GOSUB 3600
2070 GOTO 2310
2080 INK 2,6:PEN 2:PRINT"11111111111111
1111111111111111111111111";
2090 PRINT"1                                     1
         1";
2100 PRINT"1
         1";
2110 PRINT"1  1111111                            1
         1";
2120 PRINT"1  1         1111111111111111
         1";
2130 PRINT"1  1     1   1                        1
         1";
2140 PRINT"1  1  1  1   1                        1
         1";
2150 PRINT"1  1  1  1   1   111111   1
         1";
2160 PRINT"1  1  1  11 11             1  1
         1";
2170 PRINT"1  1  1      1             1  1
         1";
2180 PRINT"1  1  1111111111111  1  1
         1";
2190 PRINT"1  1                         1  1
         1";
2200 PRINT"1  1                         1  1
         1";
2210 PRINT"1  1111111111111111111   1
         1";
2220 PRINT"1                                  1
         1";
2230 PRINT"1                                  1
         1";
2240 PRINT"1111111111111111111111111111111
1111111  11";
2250 PRINT"1
         1";
2260 PRINT"1
         1";
2270 PRINT"1111111111111111111111111111111
11111111111";
2280 LOCATE 12,7:INK 3,12,22:PEN 3:IF I=
0 THEN PRINT"m" ELSE LOCATE 1,18:PRINT C
HR$(247):PRINT CHR$(247)
```

berinto con el GOSUB 2080 y se colocan las coordenadas de nuestro muñeco en la entrada. I=O significa que todavía no tenemos el disco.

—Líneas 1720-1810. Bucle principal de la fase 3. Se lee el teclado y se actualizan las coordenadas de nuestro protagonista. Se borran los enemigos, se mueven, y se vuelven a imprimir. Se comprueba si chocamos con ellos o con los ladrillos. En el primer caso se nos resta una vida (GOSUB 3410). Y en el segundo volvemos a la posición anterior guardada en las variables XB e YB.

Si cogemos un disco suena un sonido y se pasa a la fase 4. Si no, se imprime nuestro muñeco entre un grupo de gráficos, cogido al azar, proporcionándole animación.

—Líneas 1820-1880. Imprime instrucciones de la fase 4 y toca de nuevo la melodía.

—Líneas 1890-1900. Se inicializa el tiempo a 250, se coloca nuestro muñeco en el lugar donde estaba el disco y se

```
2290 FOR A=1 TO 30:LOCATE 27+12*RND,2+RN
D*14:PRINT CHR$(230);:NEXT A
2300 |SQUELCH:RETURN
2310 T=1:T1=1:S=9:B=12:D=0:D1=0:D2=0
2320 GOSUB 2580:INK 7,24:INK 2,6:INK 6,9
2330 T=T+1:IF T=6 THEN B=B+(1 AND B<S+0.
5)-(1 AND B>S+0.5):T=1
2340 PEN 6:LOCATE B-1,15:PRINT" n "
2350 S=S+(INKEY(8)=0)-(INKEY(1)=0):IF S<
2 THEN S=2 ELSE IF S>17 THEN S=17
2360 PEN 2:LOCATE S-1,24:PRINT" ef ";:PE
N 1:LOCATE S-1,25:PRINT" jk ";
2370 IF NOT INKEY(47) THEN IF D=0 THEN |
ZAP:X=23:Y=S:D=1 ELSE IF D1=0 THEN |ZAP:
X1=23:Y1=S+1:D1=1
2380 PEN 7:IF D=0 THEN 2440
2390 LOCATE Y,X:PRINT" ":X=X-1:IF X<3 TH
EN D=0:GOTO 2440
2400 LOCATE Y,X:CALL 30000:C=PEEK(29999)
:LOCATE Y,X:PRINT"o":IF C=231 THEN B=X:C
=Y:GOTO 2670
2410 IF C=110 THEN B=2+INT(RND*16)
2420 IF C>32 THEN |EXPLODE:LOCATE Y,X:PR
INT"w":P=P+15:D=0:FOR A=1 TO 100:NEXT A:
LOCATE Y,X:PRINT" "
2430 IF C>128 AND C<132 THEN PEN 5:LOCAT
E Y,X:PRINT CHR$(C-1)
2440 IF D1=0 THEN 2500
2450 LOCATE Y1,X1:PRINT" ":X1=X1-1:IF X1
<3 THEN D1=0:GOTO 2500
2460 LOCATE Y1,X1:CALL 30000:C=PEEK(2999
9):LOCATE Y1,X1:PRINT"o":IF C=231 THEN B
=X1:C=Y1:GOTO 2670
2470 IF C=110 THEN B=2+INT(RND*16)
2480 IF C>32 THEN |EXPLODE:LOCATE Y1,X1:
PRINT"w":P=P+15:D1=0:FOR A=1 TO 100:NEXT
 A:LOCATE Y1,X1:PRINT" "
2490 IF C>128 AND C<132 THEN PEN 5:LOCAT
E Y1,X1:PRINT CHR$(C-1)
2500 T1=T1+1:IF T1=15 THEN T1=1:GOSUB 25
20:IF I=1 THEN GOSUB 3280:GOSUB 3410:GOT
O 2310
2510 GOTO 2330
2520 PRINT CHR$(23);CHR$(1);:MOVE B*32-1
6,160,4:DRAWR 0,-160:FOR A=100 TO 200 ST
EP 10:SOUND 4,A,1,15:NEXT A
2530 I=0:IF B=S OR B=S+1 THEN X=23+0.5:Y
=S-0.5:I=1
2540 MOVE B*32-16,160:DRAWR 0,-160:PRINT
 CHR$(23);CHR$(0);:RETURN
2550 PRINT TAB (20-LEN(A$)/2);A$:PRINT:R
ETURN
2560 PRINT TAB(10-LEN(A$)/2);A$:PRINT:RE
TURN
```

imprime el marcador de tiempo.

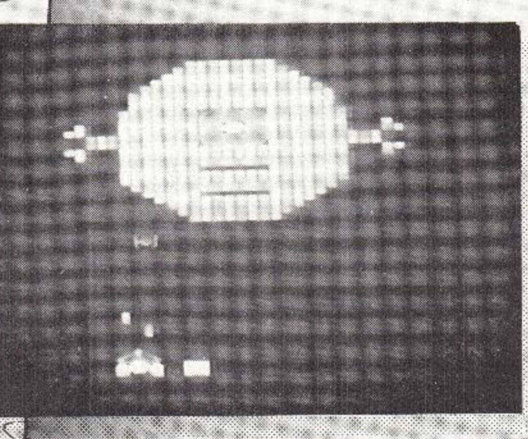
—Líneas 1910-1990. Bucle principal de la fase 4. Se cambian las coordenadas de nuestro hombre dependiendo de las teclas que pulsemos. Se comprueba si chocamos contra los ladrillos, para volvernos a la posición anterior, contra las minas, para restarnos una vida, o contra la puerta para sumarnos puntos y pasar a la fase 5.

—Líneas 2000-2070. Se imprimen las instrucciones de la fase 5.

—Líneas 2310-2320. Inicializa las variables y se dibuja la nave enemiga.

—Línea 2330-2510. Bucle principal de la fase 5. Lo primero que se hace es comprobar que cada 6 pasos del bucle, la nave escolta se mueve para situarse sobre nuestra vertical. Seguidamente se imprime. Luego se lee el teclado para comprobar si nos movemos lateralmente. Se imprime nuestra nave.

A continuación se comprue-



```
2570 PRINT:PRINT:RETURN
2580 MODE 0:INK 4,20:PEN 4
2590 A$="qpppr                        qppppppr
     qppppppppr                      pppppppppp
    sppppppppppt               q sppppppppppt r
   ppppppppppppppp             u sppppppppppt v
    sppppppppppt                 sppppppppppt
     uppppppppv                    upppppv"
2600 A$="           "+A$+"
v"                                          uppp
2610 PRINT A$
2620 PEN 3:LOCATE 9,5:PRINT"x}y":LOCATE
9,6:PRINT"} }":LOCATE 9,7:PRINT"z}{"
2630 INK 1,26:PEN 1:LOCATE 10,6:PRINT CH
R$(231)
2640 INK 5,16:PEN 5:LOCATE 9,9:PRINT CHR
$(131);CHR$(131);CHR$(131)
2650 LOCATE 9,11:PRINT CHR$(131);CHR$(13
1);CHR$(131)
2660 RETURN
2670 INK 0,0,12:BORDER 0,12:B=B-1.5:C=C-
1:FOR D=1 TO 4 STEP 3:X=B+RND*D:Y=C+RND*
D:GOSUB 3280:X=B-RND*D:Y=C-RND*D:GOSUB 3
280
2680 X=B-RND*D:Y=C+RND*D:GOSUB 3280:X=B+
RND*D:Y=C-RND*D:GOSUB 3280:NEXT D
2690 INK 0,0:BORDER 0
2700 P=P+50000:GOSUB 3420:MODE 1:LOCATE
1,5:A$="!!! LO CONSEGUISTE !!!":GOSUB 25
50:GOSUB 2570:GOSUB 2570:A$="OTRO ESCUAD
RON TE ESPERA":GOSUB 2550:A$="PARA DESTR
UIR LA SIGUIENTE":GOSUB 2550:A$="ESTRELL
A DE LA MUERTE":GOSUB 2550:GOSUB 2570:GO
SUB 2570
2710 A$="PULSA -S- PARA CONTINUAR":GOSUB
 2550
2720 IF INKEY(60)=0 THEN GOTO 1100
2730 GOTO 2720
2740 REM
2750 REM **********************
2760 REM * RUTINAS SONIDO RSX *
2770 REM **********************
2780 REM
2790 DATA 205,96,187,50,47,117,201
2800 RESTORE 2790:FOR A=30000 TO 30006:R
EAD B:POKE A,B:NEXT A
2810 RESTORE 2850:D=40000
2820 READ A$:IF A$="FIN" THEN 2850
2830 FOR A=1 TO LEN(A$) STEP 2:POKE D,VA
L("&"+MID$(A$,A,2)):D=D+1:NEXT A
2840 GOTO 2820
```

ba si pulsamos la tecla de disparo. Si la pulsamos y éste no existe, el disparo 1 se activa. Si hay disparo, se borra, se mueve hacia arriba y si sobrepasa el límite, se desactiva. Se comprueba si choca contra la nave escolta, de ser así ésta toma una nueva posición al azar. Si choca contra la nave enemiga, la va destruyendo (si son escudos necesita varios disparos), y si da al ojo energético salta a la rutina de explosión de la nave enemiga.

Estas mismas operaciones se repiten con el disparo de la derecha.

—Línea 2520-2540. Subrutina de disparo de la nave escolta. Si nos alcanza explota nuestra nave y se nos resta 1 vida, de lo contrario continuamos de nuevo en el bucle principal.

A continuación se detalla la función que cumple cada subrutina.

—Línea 2550. Imprime en

```
2850 DATA 014A9C215E9CCDD1BCC9629CC3829C
C3A19CC3C09CC3DF9CC31B9DC34C9D000000045
58504C4F44C5535155454C43C850494EC75A41D0
414C4945CE53484F4FD400CDA7BC3E0121949CCD
BCBC21989CCDAABCC9010FFF1901010000000F0F
0000CDA7BC060FC57832BC9C21B79CCDAABC30FB
C110F0C901
2860 DATA 010000000000F0200CDA7BC3E0121D2
9CCDBCBC21D69CCDAABCC9010FFF050101001900
000F0000CDA7BC3E0121059DCDBFBC21099DCDAA
BC060FC53E109032179D21129DCDAABC30FBC110
EEC9010F0402010001280000061E000200000000
000F0200CDA7BC3EFF21339DCDBFBC213A9DCDAA
BC21439DCD
2870 DATA AABCC90205FF020501020100016400
0007640002000016C0000076400CDA7BC3E012166
9DCDBCBC216A9DCDAABC21739DCDAABC30FBC901
0FFF020100000000A0F040001010000000050F00
00
2880 DATA FIN
2890 REM
2900 REM **********************
2910 REM * REDEFINIR GRAFICOS *
2920 REM **********************
2930 REM
2935 SYMBOL AFTER 96
2940 SYMBOL 97,0,0,3,62,255,28,3,0
2950 SYMBOL 98,14,18,226,66,254,2,193,63
2960 SYMBOL 99,6,7,13,27,49,96,207,248
2970 SYMBOL 100,0,0,192,176,76,115,94,24
0
2980 SYMBOL 101,1,3,5,25,50,101,194,127
2990 SYMBOL 102,128,192,160,152,76,166,6
7,254
3000 SYMBOL 103,0,0,3,13,50,206,122,15
3010 SYMBOL 104,96,224,176,216,140,6,243
,31
3020 SYMBOL 105,60,126,255,127,255,254,1
27,28
3030 SYMBOL 106,3,103,235,247,203,211,24
0,240
3040 SYMBOL 107,192,230,215,239,211,203,
15,15
3050 SYMBOL 108,0,239,239,239,0,254,254,
254
3060 SYMBOL 109,255,255,255,231,231,255,
252,255
3070 SYMBOL 110,129,153,189,231,189,153,
129,66
3080 SYMBOL 111,0,24,24,24,24,24,24,0
3090 SYMBOL 112,238,238,238,0,238,238,23
8,0
```

MODO 1, centrada, la frase contenida en la variable A$.

—Línea 2560. Igual que la anterior por el MODO 0 de pantalla.

—Líneas 2580-2660. Se imprime la nave mala, que sale en la fase 5.

—Líneas 2660-2730. Explosión de la nave enemiga. Bono de 50.000 y vuelve a comenzarse el juego.

—Línea 2740-2880. Se introducen en memoria dos subrutinas en C/M. La primera se encarga de reconocer el carácter de la pantalla, que hay en la posición del cursor, que es guardado en la posición de memoria 29999, y la segunda, es una pequeña ampliación de comandos RSX que nos permiten generar los sonidos de explosión, campanas, lasers, etc.

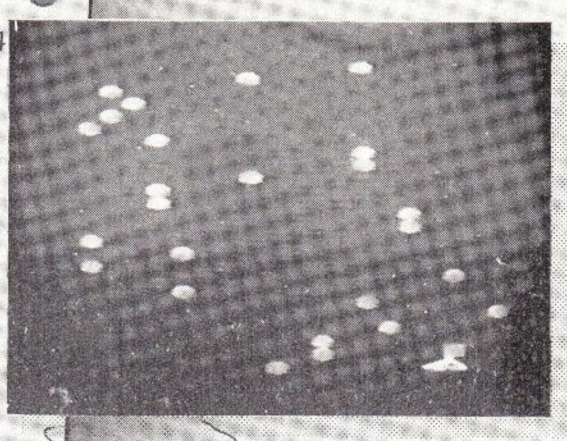
—Líneas 2890-3210. Se redefinen los gráficos.

—Línea 3220. Se activan los comandos RSX.

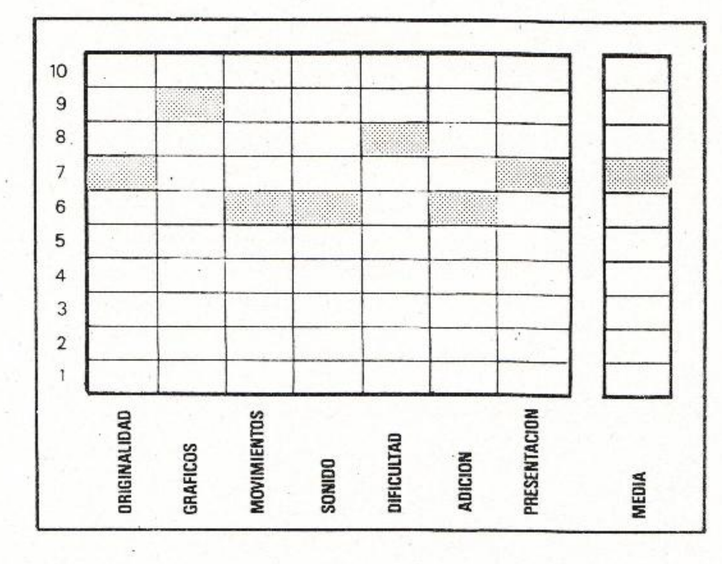
—Líneas 3230-3350. Subrutina de explosión.

—Líneas 3360-3440. Se res-

(Continúa en pág. 21)



MERMAID MADNESS



VIDEO-AVENTURA

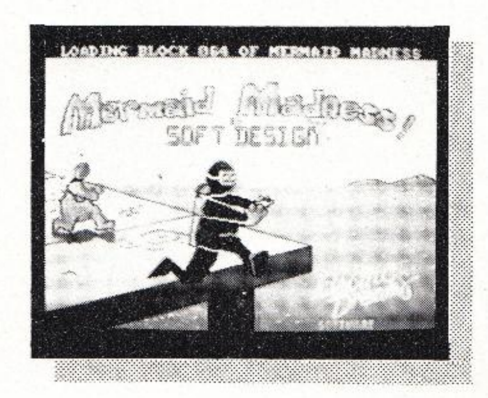
ELECTRIC DREAMS SOFTWARE

Nada hacia las profundidades del mar con Myrtle para rescatar a su amado Gordon. Para ello deberás sortear las criaturas marinas que te vayas encontrando, ya que te atacarán con tan solo rozarlas. Tendrás que beber cerveza negra para mantener alto el nivel de energía. Deberás recoger también todos los objetos que encuentres en tu camino.

En esta comedia arcade toma parte el submarinista Gormless Gordon que ha sido asaltado por la sirena Myrtle cuyo rostro ha conseguido hundir varios miles de barcos.

Emocionante música, estremecimientos en el fondo. Atrévete a llegar donde nadie ha sido capaz, y descubre los tesoros escondidos que harán flaquear tus rodillas.

La carga de la aventura, se realiza mediante una rutina especial, e incluye una cuenta atrás de bloques que amenizan la espera. La pantalla de presentación ha sido realizada en el MODO 1 de pan-

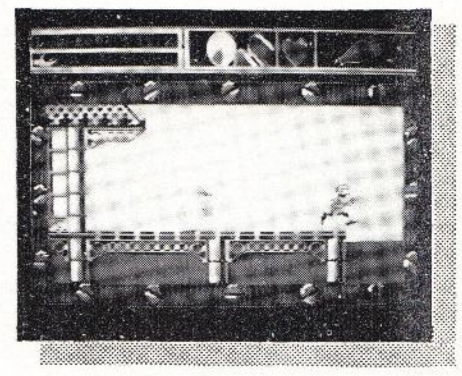


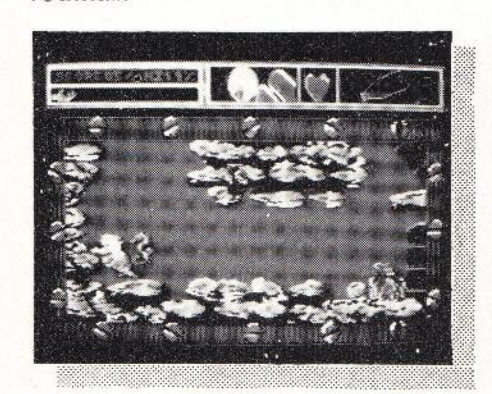
talla, pero han conseguido aprovechar muy bien los cuatro colores.

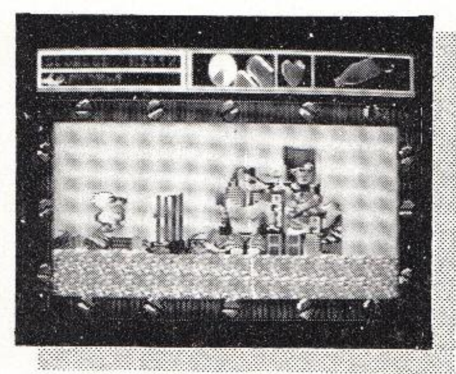
Al comenzar el juego, lo primero que nos llama la atención es la melodía que le acompaña, que se hace muy pegadiza.

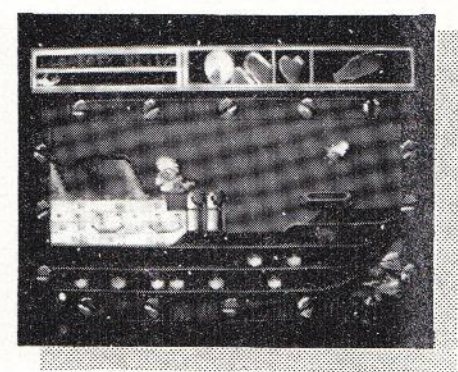
Por la vista, unas pantallas de agradables gráficos nos inundan, y comprobamos como por interrupciones han logrado mantener la zona de los marcadores en el MODO 1 de pantalla aprovechando así la mayor resolución, y el resto de la pantalla en el MODO 0 para disponer así de 16 colores.

Pero estas cualidades se ven mermadas por la lentitud de los movimientos y por las dificultades que nos presentan los animales marinos, que a su roze entorpecen mucho el desarrollo del juego.









NONAMED

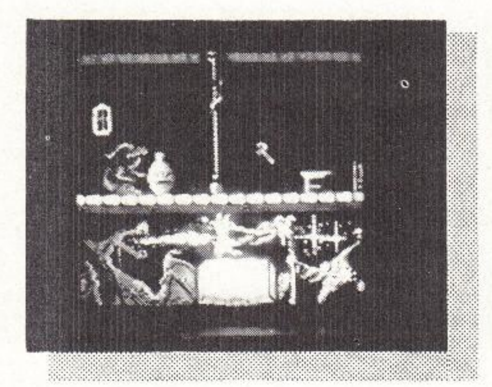
VIDEO-AVENTURA *DINAMIC*

Erase que se era una época pasada donde los hombres para demostrar su valor y conseguir ser nombrados Caballeros del Rey, debían someterse a las más duras pruebas.

En un valle olvidado de la Baja Sajonia existía un pueblo de guerreros y campesinos que rendía pleitesía a su anciano y venerable monarca, el rey ABDUL HONEICKAN GARGOY.

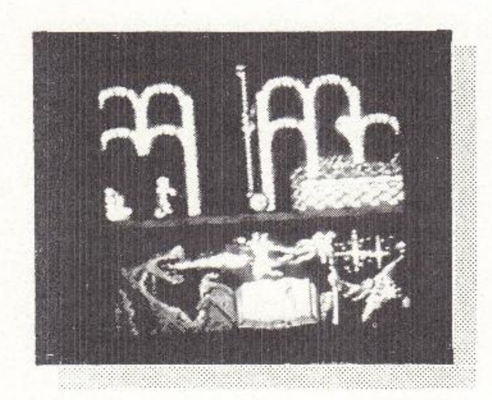
Los jóvenes aspirantes a Caballeros de su Majestad eran seleccionados en todas las comarcas y villas de entre aquellos que destacaban en el manejo del arco, la lucha a cuerpo o la espada.

Sin embargo, el más duro de los sacrificios quedaba reservado sólo a unos pocos elegidos.



Uno por uno eran sometidos al rito final, el rito del Castillo sin nombre.

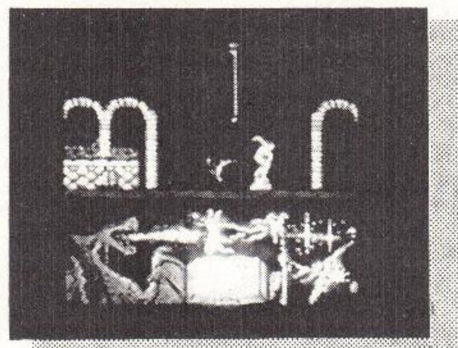
Consistía la prueba en ser encerrado en el Castillo sin nombre, intentar sobrevivir al mundo de pesadilla que existía dentro de sus muros y escapar con vida. Para ello la cuestión fundamental era encontrar la puerta.



Muchos jóvenes aspirantes a caballeros intentaron el reto, pero la gran mayoría nunca consiguieron sobrevivir y fueron convertidos en ogro bajo los poderosos hechizos que el malvado mago NILREM reservaba a los perdedores.

Para escapar con vida del castillo sin nombre, deberás entregar algo al mago NILREM que en agradecimiento te pagará con un conjuro.

Deberás impedir el contacto con los ogros, en caso contrario tu suerte será fatal.

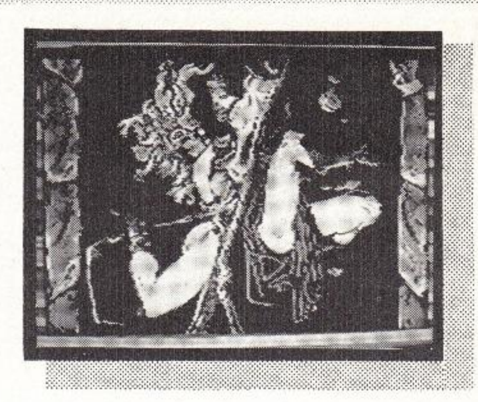


Otra de las claves del éxito tiene relación con los jardines interiores del castillo en donde se encuentra la Fuente Sagrada, su agua te dará mucha fuerza y hara más vigorosos tus saltos.

Las capacidades físicas del protagonista son variadas, puede correr para escapar de sus perseguidores, golpearles a base de patadas, trepar por cuerdas y agacharse para no ser herido por los enemigos aéreos.

Los movimientos tanto de nuestro personaje como los de los enemigos son por lo general rápidos y van combinados con un scroll mediante el cual se desplaza el paisaje lateralmente.

Para ascender o descender a otras plantas, disponemos de cuerdas por las que trepar o deslizarnos.



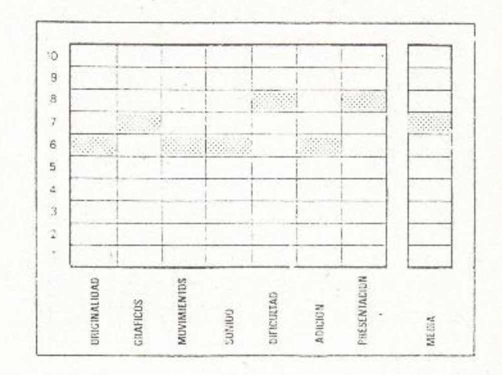
En lo referente al sonido, una melodía acorde con el tema, acompaña el menú de opciones. En cambio, durante el juego el sonido es escaso. Se hubiese agradecido la presencia de otros sonidos además de el de las pisadas de nuestro personaje.

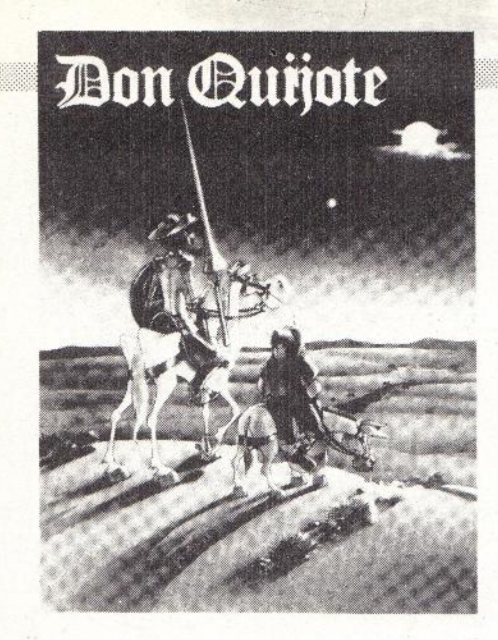
La realización de los gráficos está muy trabajada (algo muy habitual en DINAMIC), de hecho el marcador de las vidas ocupa "un tercio de la pantalla". Decimos esto a modo anecdótico para resaltar las exuberancia gráfica que han empleado, como veréis en las fotografías de la revista.

Existen diversos tipos de enemigos, que tenemos que combatir de formas variadas. A unos tendremos que eliminarlos a golpes (nuestro protagonista tiene dotes de karateka), como a los ogros y esqueletos, a otros solo podremos esquivarles agachándonos o saltando como por ejemplo, el águila, las hachas, bolas que ruedan por el suelo, etc...

La dificultad es elevada, no hemos acabado de eliminar o esquivar a un enemigo cuando otro se avalanza ya sobre nosotros. Pero el deseo del jugador por conocer nuevas pantallas hacen de NONAMED un juego adictivo.

La idea de la aventura no es muy original, pero la presentación ha sido cuidada en todos sus detalles.





DON QUIJOTE DE LA MANCHA

AVENTURA GRAFICO-CONVERSACIONAL

DINAMIC

INTRODUCCION:

La Mancha, España, abril de 1605...

Eres don Alonso Quijano, un hidalgo manchago del siglo XVI, cuya debilidad son las novelas de caballerías. Todo va más o menos bien por tu casa, hasta que un día tu locura alcanza el punto culminante: decides coger las armas y salir a correr aventuras por la Mancha. ¡Vaya ideas extrañas que tiene el hidalgo! Te consigues un escudero, Sancho Panza, que, aunque no es gran cosa, te ayudará en los momentos difíciles. A partir de este momento entrarás en las más increíbles aventuras que te puedas imaginar. Ventas, colinas, bosques y posadas serán testigos de tus más locas ideas tales como atacar molinos o enfrentarte a fieros leones.

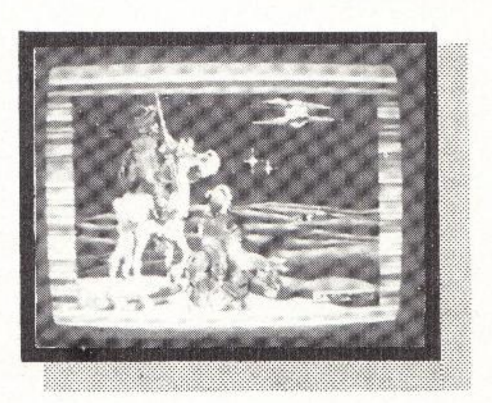
Tu objetivo final es conseguir los amores del Dulcinea del Toboso, tu dama. Aunque existe un primer objetivo, que es armarte caballero, para poder así comenzar tus desventuras.



COMENTARIO:

Don Quijote es un juego que pone a prueba tu ingenio y tu capacidad de observación, ya que en él tendrás que resolver muchas situaciones embarazosas, sin más ayuda que la de tu memoria y las pistas que el programa te va dando ocultas o, disfrazadas a veces, entre el texto o diálogo que el ordenador mantiene contigo.

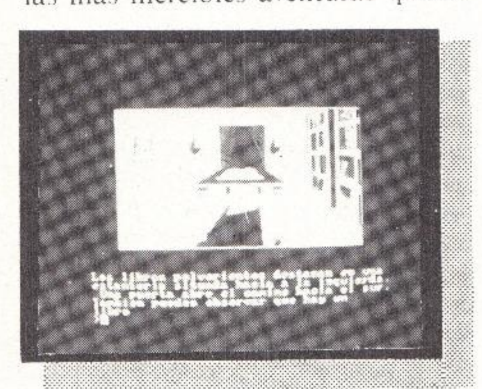
Don Quijote no es en absoluto un juego convencional y tampoco se asemeja a las aventuras gráfico-conversacionales clásicas, donde la respuestas del jugador debían ajustarse a un formato rígido y los gráficos eran pocos o de escasa calidad.

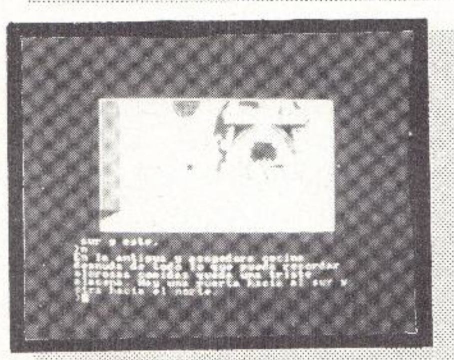


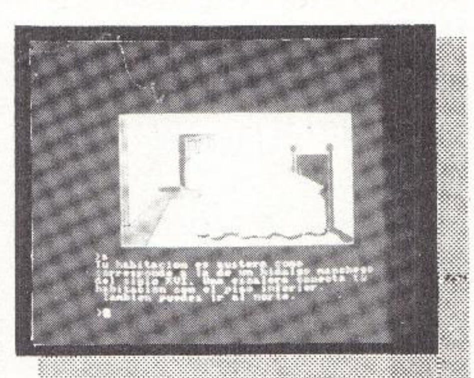
En Don Quijote por el contrario, te encontrarás con que el ordenador comprende frases ligeramente complicadas, y te sorprenderá la gran calidad de los dibujos que aparecen en cada situación. Sin embargo quizá se hechen en falta algunos sonidos ambientales, que hubieran dado mayor emoción al juego.

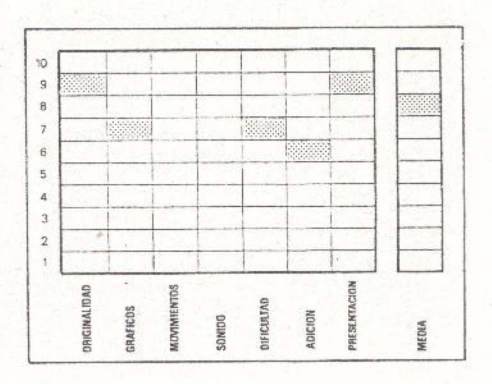
Como pantalla de presentación tenemos un estupendo dibujo, en el que aparecen Don Quijote y Sancho sobre sus respectivas cabalgaduras, en un paisaje típico manchego.

Como ya hemos dicho, Don Quijote no puede ser comparado con ningún juego convencional ya que su temática es totalmente distinta, sin embargo podemos asegurar sin temor a equivocarnos que es la mayor aventura gráfico-conversacional que se ha realizado en lengua española.

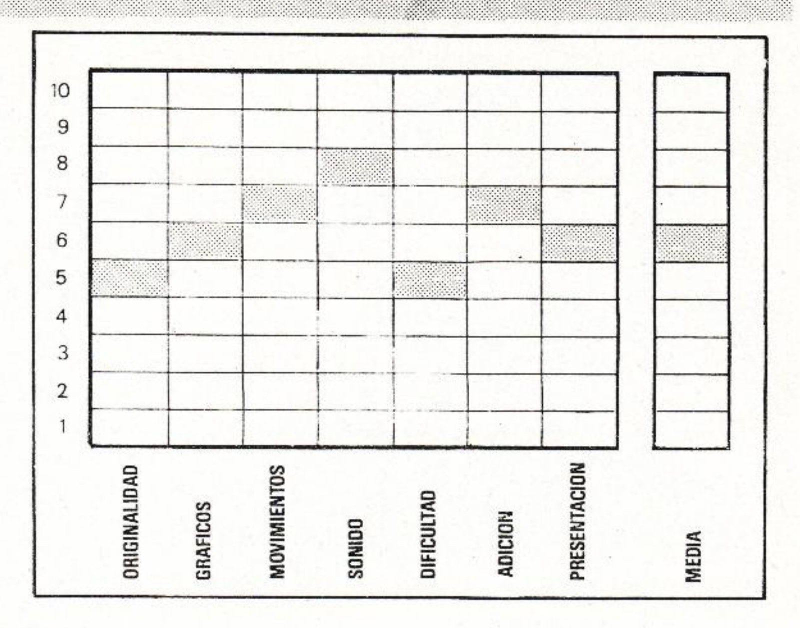








VIERNES 13



FRIDAY THE 13

VIDEO-AVENTURA

ZCOBRA

Antes de empezar a jugar asegúrate de:

1. Cerrar todas las puertas, ventanas y cortinas.
2. Apagar todas las luces y utilizar una vela si es necesario.
3. Comprobar que GRANNY no está en el cuarto.
4. Poner el volumen al máximo. Entonces carga y juega.

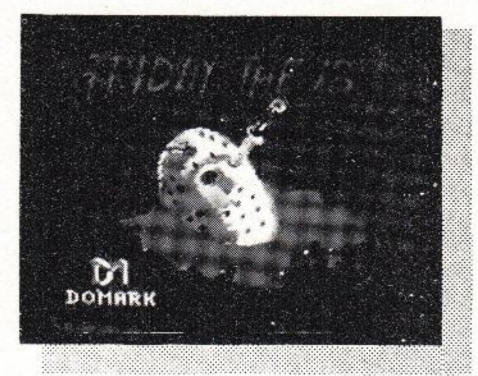
Jason está escondido en el bosque, esperando vengar la muerte de su madre, cuando un grupo de adolescentes en vacaciones visitan el campamento del misterioso CRISTAL LAKE cercano a una gruta...

Jason se pone su máscara de hockey y afila su machete preparado para una masacre desbordada.

Debes avisar a todos que el asesino loco está al acecho y conducirles a la salvación, por supuesto sin perder tu cabeza.

COMENTARIO

La idea del juego ha sido tomada del film que lleva el mismo título. Aún así muestra un ligero toque de originalidad. Durante la carga, tras mostrarnos la sangrienta pantalla de presentación, nos sorprendió un grito de agonía que asustará a más de uno.



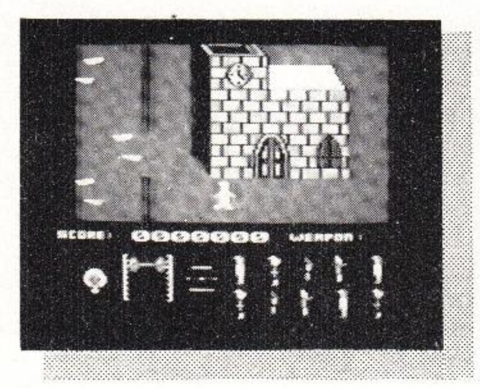
El programa posee muchas pantallas, bien ambientadas, y con un gran derroche de gráficos. También presenta un efecto de tridimensionalidad, acompañado de la posibilidad de pasar por delante o por detrás de los objetos.

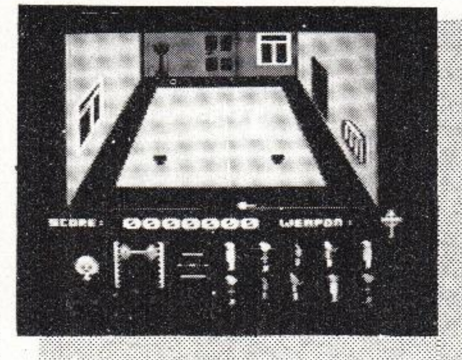
Distribuidas por todo el escenario existen numerosas armas, unas arrojadizas y otras que no lo son, como: cuchillos, hachas, sierras mecánicas, lanzas, palos, tridentes, etc...

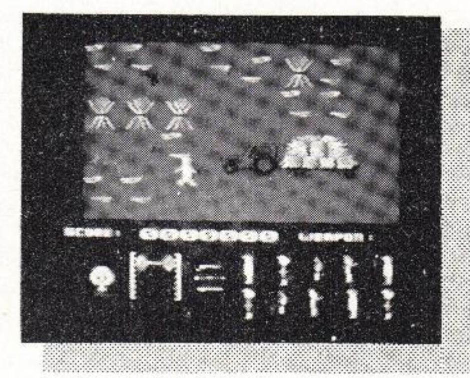
Los gráficos aunque muy variados, no son de gran calidad, tal vez debido a su pequeño tamaño.

El sonido en todo momento acompaña a la acción, distintas melodías suenan mientras intentamos encontrar a Jason. Cuando éste elimina a una víctima oímos su grito de horror, y en los marcadores desaparece la figura de la víctima, siendo sustituida por una lápida mortuoria.

Cuando empieces a jugar te será difícil despegarte de tu ordenador. El momento que reviste más emoción, es cuando encontramos a Jason, comenzando aquí una encarnizada persecución que terminará con un sólo vencedor. Salvo esta situación el resto del juego no reviste dificultad.







```
3100 SYMBOL 113,14,14,14,0,238,238,238,0
3110 SYMBOL 114,224,224,224,0,238,238,23
8,0
3120 SYMBOL 115,14,14,14,0,14,14,14,0
3130 SYMBOL 116,224,224,224,0,224,224,22
4,0
3140 SYMBOL 117,238,238,238,0,14,14,14,0
3150 SYMBOL 118,238,238,238,0,224,224,22
4,0
3160 SYMBOL 119,8,65,16,130,40,129,8,34
3170 SYMBOL 120,15,51,103,79,223,255,255
,255
3180 SYMBOL 121,240,252,254,254,255,255,
255,255
3190 SYMBOL 122,255,255,255,255,127,127,
63,15
3200 SYMBOL 123,255,255,255,255,254,254,
252,240
3210 SYMBOL 125,255,255,255,255,255,255,
255,255
3220 CALL 40000:RETURN
3230 REM
3240 REM *************
3250 REM * EXPLOSION *
3260 REM *************
3270 REM
3280 YB=399-X*16
3290 X=Y*32:Y=YB
3300 ORIGIN X,Y:TAG
3310 |EXPLODE:FOR A=10 TO 80 STEP 2
3320 MOVE A*RND,A*RND,1:PRINT"w";:MOVE -
A*RND,-A*RND:PRINT"w";
3330 MOVE -A*RND,A*RND:PRINT"w";:MOVE A*
RND,-A*RND:PRINT"w";
3340 NEXT A:FOR A=1 TO 200:NEXT A:TAGOFF
3350 RETURN
3360 REM
3370 REM **************
3380 REM * RESTA VIDA *
3390 REM **************
3400 REM
3410 VI=VI-1
3420 MODE 1:PEN 1:LOCATE 1,10:IF VI<1 TH
EN 3500
3430 A$="PUNTOS:"+STR$(P):GOSUB 2550:A$=
" ":GOSUB 2550:A$="VIDAS:"+STR$(VI):GOSU
B 2550
3440 FOR A=1 TO 2500:NEXT A:MODE 0:PRINT
 CHR$(7);:RETURN
3450 REM
3460 REM *************
3470 REM * GAME OVER *
3480 REM *************
```

ta una vida, si no nos quedan vidas se salta a GAME OVER.

—Líneas 3450-3540. Se han agotado las vidas. Mensaje de GAME OVER. Pregunta si deseas jugar otra partida. Contestando S se vuelve a jugar. Contestando N el programa se despide y se autodestruye.

—Líneas 3550-3670. Melodía que suena entre fase y fase.

SOLO PARA ADICTOS

—Para conseguir vidas infinitas, sustituir la líneas 3410 por:

3410 VI=VI-O

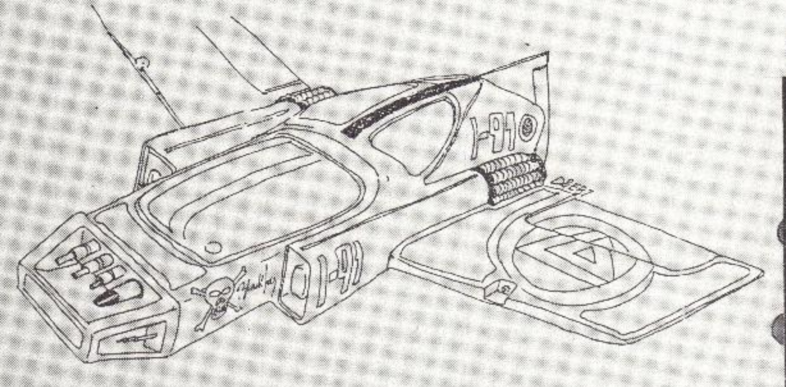
—Para aumentar el tiempo que se tiene para escapar en la fase 4, cambiar el valor dado a T por uno mayor. Por ejemplo:

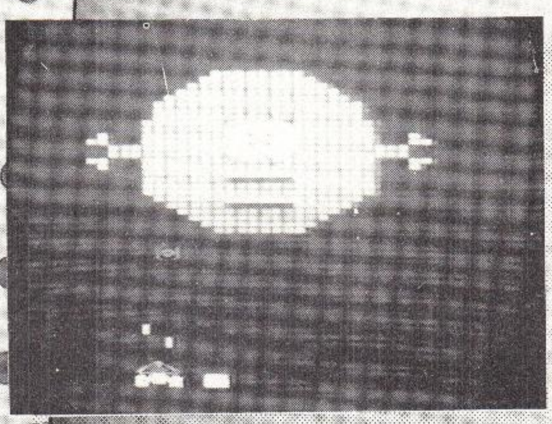
1890 T=300

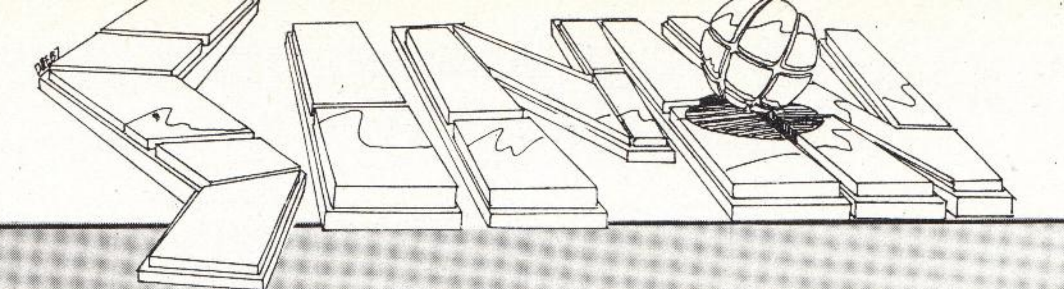
```
3490 REM
3500 LOCATE 1,5:A$="GAME OVER":GOSUB 255
0:GOSUB 2570:A$="TU ESCUADRON HA FRACASA
DO":GOSUB 2550:A$="EN SU INTENTO DE ACAB
AR":GOSUB 2550:A$="CON LAS FUERZAS DEL M
AL":GOSUB 2550:GOSUB 2570
3510 GOSUB 2570:A$="TU PUNTUACION:"+STR$
(P):GOSUB 2550:A$="OTRA PARTIDA (S/N)":G
OSUB 2550
3520 IF NOT INKEY(60) THEN |PING:RUN 106
0
3530 IF NOT INKEY(46) THEN MODE 0:LOCATE
 1,12:A$="!!! ADIOS !!!":GOSUB 2560:|PIN
G:FOR A=1 TO 2500:NEXT A:CALL 0
3540 GOTO 3520
3550 REM
3560 REM **********
3570 REM * MUSICA *
3580 REM **********
3590 REM
3600 ENT -1,8,1,1,16,-1,1,16,1,1,8,-1,1
3610 SOUND 1,536,100,7,0,1:SOUND 2,268,1
00,7,0,1:SOUND 4,0,300,0
3620 SOUND 1,478,100,7,0,1:SOUND 2,239,1
00,7,0,1
3630 SOUND 1,602,200,7,0,1:SOUND 2,301,2
00,7,0,1
3640 SOUND 1,602,200,6,0,1:SOUND 2,301,2
00,6,0,1
3650 SOUND 4,1204,100,15,0,1:SOUND 4,804
,200,15,0,1
3660 WHILE SQ(1)>128:WEND
3670 RETURN
```

—Para modificar la duración de la fase 2, es necesario cambiar el valor de T en la línea 1530.

- Para hacer que sea mayor T, debe tomar un valor entre 0 y 250.
- Y para que se acorte T, debe estar entre 250 y 480.







SIMON.— Una de las cualidades que dotan de interés a un juego, es el progresivo aumento de la dificultad a medida que se avanza.

Simon, es un juego que cumple perfectamente con este objetivo. El ordenador genera aleatoriamente una secuencia de luces y sonidos, que el jugador ha de repetir en el mismo orden. Si se aciertan todas, el ordenador añade una secuencia más al conjunto, aumentando como decíamos el nivel de dificultad.

```
1000 REM *************************
1010 REM * ANGEL GARCIA DELGADO *
1020 REM *************************
1030 REM
1040 REM *********
1050 REM * SIMON *
1060 REM *********
1070 REM
1080 REM
1090 REM ****************
1100 REM * PRESENTACION *
1110 REM ****************
1120 REM
1130 MODE 1:INK 0,0:PAPER 0:BORDER 0
1140 INK 1,6:INK 2,26
1150 PEN 1
1160 LOCATE 1,6
1170 PRINT"                   #### ### ##### #### #
###"
1180 PRINT"                   #    #  # # # #  # #
 #"
1190 PRINT"                   ####  #  # # # #  # #
 #"
1200 PRINT"                      #  #  # # # #  # #
 #"
1210 PRINT"                   #### ### # # # ####  #
 #"
1220 PRINT
1230 PEN 2
1240 PRINT
1250 PRINT"              <<< MUNDO AMSTRAD >
>>"
1260 PRINT
1270 PRINT"                 1987 UNICORNIO SOF
T"
1280 PRINT
1290 PRINT
1300 PRINT"          POR: ANGEL GARCIA DEL
GADO"
1310 FOR A=1 TO 250
1320 A$=INKEY$
1330 NEXT A
```

Sin duda, un juego sencillo, pero divertido. Los controles son:

1. Luz num. 1
2. Luz num. 2
3. Luz num. 3
4. Luz num. 4

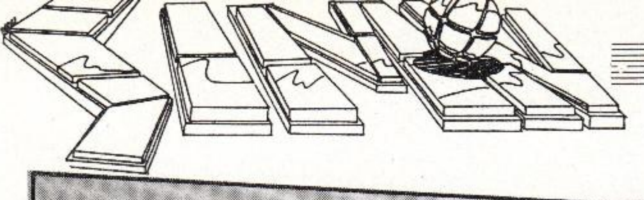
Analicemos como está hecho:

—Líneas 1000-1350.— Se dibuja la pantalla de presentación. Se fija el MODO 1 como modo de pantalla, se definen los colores para las eintas, el papel y el borde, y con PRINT y LOCATE se imprimen los rótulos.

En las líneas 1310-1330 se limpia el buffer del teclado y en las siguientes se genera un sonido al azar mientras se aguarda la pulsación de una tecla.

En las líneas 1360-1550. Se dibuja el Simon. Se activa el MODO 0. Y mediante 2 bucles, y el empleo de las funciones trigonométricas SIN y COS se dibuja, primero los sectores del interior, y luego el círculo que los rodea. A continuación en la línea 1530 activamos el modo





```
1340 SOUND 1,RND*50,1,15:INK 1,RND*26
1350 IF INKEY$="" THEN 1340
1360 REM
1370 REM ******************
1380 REM * DIBUJO PANTALLA *
1390 REM ******************
1400 REM
1410 MODE 0:INK 0,0:PAPER 0:BORDER 0:INK
 5,26:GOSUB 1850
1420 ORIGIN 320,200
1430 FOR a=0 TO PI/2 STEP PI/150
1440 c=COS(a)*100:s=SIN(a)*100
1450 PLOT 10,6,2:DRAWR c,s
1460 PLOT -10,6,1:DRAWR -c,s
1470 PLOT -10,-6,3:DRAWR -c,-s
1480 PLOT 10,-6,4:DRAWR c,-s
1490 NEXT a
1500 FOR A=0 TO 2*PI STEP PI/30
1510 MOVE 130*(COS(A-PI/30)),120*(SIN(A-
PI/30)),6:DRAW 130*COS(A),120*SIN(A)
1520 NEXT A
1530 PEN 5:PRINT CHR$(22);CHR$(1)
1540 LOCATE 9,10:PRINT"1  2":LOCATE 9,16
:PRINT"3  4"
1550 PRINT CHR$(22);CHR$(0)
1560 REM
1570 REM *****************
1580 REM * COMIENZO JUEGO *
1590 REM *****************
1600 REM
1610 A$="":GOSUB 1920:C$=""
1620 A$="ATENTO EMPEZAMOS":GOSUB 1920
1630 FOR A=1 TO 1000 STEP 20:SOUND 1,A,1
,15:SOUND 2,1000-A,1,15:NEXT
1640 FOR A=1 TO 1000:NEXT
1650 C$=C$+CHR$(49+INT(RND*4))
1660 FOR B=1 TO LEN(C$):A$=MID$(C$,B,1):
GOSUB 1990:FOR Y=1 TO 100:NEXT Y
1670 NEXT B
1680 A$="ES TU TURNO":GOSUB 1920
1690 FOR D=1 TO LEN(C$):A$=MID$(C$,D,1)
1700 R$=""
1710 IF NOT INKEY(64) THEN R$="1"
1720 IF NOT INKEY(65) THEN R$="2"
1730 IF NOT INKEY(57) THEN R$="3"
1740 IF NOT INKEY(56) THEN R$="4"
1750 IF R$="" THEN 1710
1760 IF R$=A$ THEN GOSUB 1990 ELSE 2100
1770 NEXT D
1780 A$="MUY BIEN":GOSUB 1920:A$="AHORA
OTRA MAS":GOSUB 1920
1790 GOTO 1650
1800 REM
1810 REM ******************
```

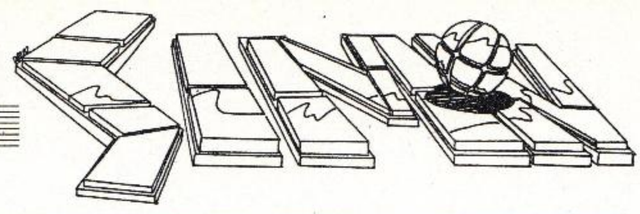
transparente, y sobreimpresionamos los números de los sectores. Luego en la 1550 lo desactivamos.

—Líneas 1560-1790. Este bloque construye el bucle principal del juego. Comienza inicializando la variable C$ a la cadena vacia, puesto que C$ va ha contener la frecuencia de números que SIMON va a ir generando. En la siguiente línea, se anuncia que el juego va ha comenzar, y tras un sonido, y una pausa, en la línea 1650 el ordenador suma a la cadena C$ un nuevo número comprendido entre 1 y 4.

En la línea 1660-70 mediante un FOR-NEXT el ordenador muestra la secuencia que el jugador tendrá que memorizar y luego repetir.

En la línea 1680 desafía al jugador para que repita la secuencia, que va a ser tomada y comprobada en las líneas siguientes 1690-1770. En este grupo de líneas, se realiza un bucle en el que se comprueba que teclas pulsa el jugador, activando la luz correspondiente al llamar a la subrutina de la línea 1990 que más adelante detallaremos. En R$ se guarda qué tecla pulsamos, y se compara con la secuencia actual guardada en A$. Si no son iguales se salta a la línea 2100, a partir de la cual se nos indica que hemos fallado. En el caso de que si sean igual se continúa con el bucle hasta finalizarlo.

Cuando se termina el bucle, quiere decir, que hemos acertado todas las secuencias, mensaje que se da en la línea 1780. A continuación se salta a la línea 1650 para añadir otro número más.



```
1820 REM * RESTAURA TINTAS *
1830 REM *******************
1840 REM
1850 PEN 6,1:INK 1,3:INK 2,9:INK 3,12:IN
K 4,10
1860 RETURN
1870 REM
1880 REM ***********
1890 REM * VENTANA *
1900 REM ***********
1910 REM
1920 PEN 5:PRINT CHR$(22);CHR$(0):LOCATE
 10-LEN(A$)/2,1:PRINT A$;:FOR W=1 TO 100
0:NEXT W:LOCATE 1,1:PRINT"
       ";
1930 RETURN
1940 REM
1950 REM ******************
1960 REM * ENCIENDE LUCES *
1970 REM ******************
1980 REM
1990 ENT 1,5,-1,1,5,1,1,10,0,1:IF A$="1"
 THEN INK 1,15:SOUND 1,75,20,15,0,1:GOTO
 2090
2000 IF A$="2" THEN INK 2,18:SOUND 1,125
,20,15,0,1:GOTO 2090
2010 IF A$="3" THEN INK 3,24:SOUND 1,175
,20,15,0,1:GOTO 2090
2020 IF A$="4" THEN INK 4,20:SOUND 1,225
,20,15,0,1:GOTO 2090
2030 RETURN
2040 REM
2050 REM *********
2060 REM * FALLO *
2070 REM *********
2080 REM
2090 FOR X=1 TO 200:NEXT X:GOTO 1850
2100 A$="!!! NOOOO !!!":SOUND 1,1400,120
,15:GOSUB 1920
2110 A$="LA SOLUCION ERA":GOSUB 1920
2120 FOR B=1 TO LEN(C$)
2130 A$=MID$(C$,B,1):GOSUB 1990
2140 FOR Y=1 TO 100:NEXT Y
2150 NEXT B
2160 PEN 5:LOCATE 1,1:PRINT"OTRA PARTIDA
 (S/N)?"
2170 FOR A=1 TO 500:NEXT
2180 IF NOT INKEY(60) THEN 1610
2190 IF NOT INKEY(46) THEN CALL 0
2200 INK 1+INT(RND*4),INT(RND*26)
2210 SOUND 1,RND*20,5,10
2220 FOR A=1 TO 10:NEXT
2230 GOSUB 1850
2240 GOTO 2180
```

—Líneas 1800-1860.— Apaga las luces.

—Líneas 1880-1930.— Subrutina de impresión de mensajes (ventana). Esta subrutina imprime el texto centrado, hace una pausa, y borra el mensaje.

—Líneas 1940-2030.— Esta es la subrutina que enciende la luz cuyo número se indica en la variable de cadena A$.

—Líneas 2040-2240.— Se salta a esta rutina cuando se comete un fallo. En la línea 2100 se dice que has fallado, y a continuación el ordenador te ofrece la solución. Por último te pregunta si deseas jugar otra vez, en caso negativo, el programa se destruye, al inicializarse el ordenador (CALL 0).

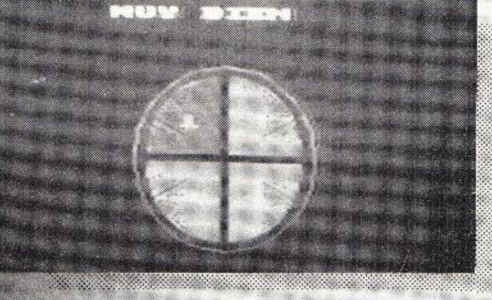
Como aspecto educativo del programa señalaremos, cómo se encienden y se apagan las luces, sólo cambiando el color de las tintas. La luz está apagada cuando tiene un tono mate (sin brillo) y da la impresión de que se enciende al poner el mismo color más claro (con brillo).

SOLO PARA ADICTOS:

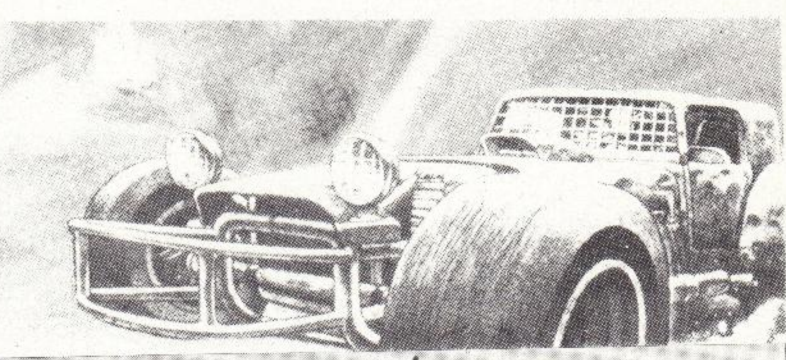
—Si no quieres fallar nunca, sustituye la línea 1760 por: 1760 GOSUB 1990

—Si quieres saber cuantos colores has acertado añade la línea:

2155 A$="acertaste:"+STR$ (LEN(C$)):GOSUB 1920



ATROPELLO



```
1000 REM ************************
1010 REM * ANGEL GARCIA DELGADO *
1020 REM ************************
1030 REM
1040 REM ************************
1050 REM *       ATROPELLO      *
1060 REM ************************
1070 REM
1080 REM
1090 REM ****************
1100 REM * PRESENTACION *
1110 REM ****************
1120 REM
1130 MODE 0
1140 INK 0,0:INK 1,26:INK 2,7:INK 3,18
1150 PAPER 0:BORDER 0
1160 GOSUB 2420
1170 PRINT:PRINT
1180 PRINT"  aaaaaaaaaaaaaaaa"
1190 PRINT"  a              a"
1200 PRINT"  a  ATROPELLO.  a"
1210 PRINT"  a              a"
1220 PRINT"  aaaaaaaaaaaaaaaa"
1230 PRINT
1240 PRINT
1250 PRINT"           1987"
1260 PRINT
1270 PRINT"  UNICORNIO SOFT"
1280 PRINT
1290 PRINT
1300 PRINT
1310 PRINT" ddddddddddddddddd"
1320 PRINT"d                 d";
1330 PRINT"d ANGEL GARCIA DGO.d";
1340 PRINT"d                 d";
1350 PRINT" ddddddddddddddddd"
1360 S=1
1370 FOR A=1 TO 50:A$=INKEY$:NEXT
1380 SOUND 1,S,1,10:S=S+5:IF S>1000 THEN
 S=0
1390 IF INKEY$<>"" THEN 1410
```

ATROPELLO.— Es un programa para dos jugadores. El primero de ellos (jugador de la izquierda), controla los movimientos del atrevido peatón. El otro jugador (jugador de la derecha), maneja un automóvil con el cual ha de atropellar al peatón, antes de que el tiempo se agote y el transeunte se escape.

Las teclas de control del peatón son:

Q. ARRIBA
A. ABAJO
X. IZQUIERDA
C. DERECHA

...y las del maniático del volante son:

Cursor arriba.	ARRIBA
Curso abajo.	ABAJO
Cursor derecha.	DERECHA
Cursor izquierda.	IZQUIERDA

La mejor manera de entender un programa es entender su funcionamiento. Por lo tanto, vamos a intentar explicar de una forma sencilla la manera en que trabaja este programa.

Visto a grandes rasgos, el programa consta de tres partes: la presentación, el juego y la conclusión.

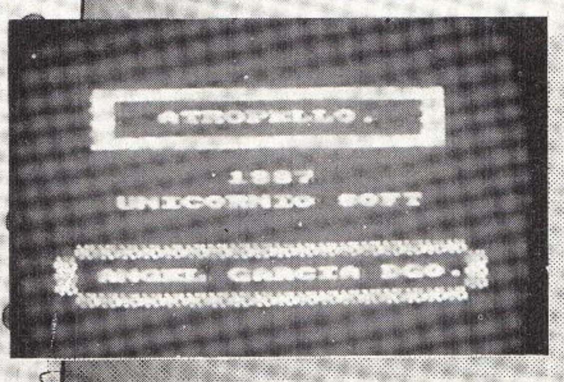
```
1400 GOTO 1380
1410 MODE 1
1420 GOTO 1470
1430 PRINT TAB(20-LEN(a$)/2):FOR a=1 TO
LEN(a$)
1440 b$=MID$(a$,a,1):PRINT b$;:IF b$<>"
" THEN SOUND 1,A,1,10
1450 NEXT a
1460 PRINT:PRINT:FOR a=1 TO 200:NEXT a:R
ETURN
1470 REM
1480 REM ****************
1490 REM * INSTRUCCIONES *
1500 REM ****************
1510 REM
1520 a$="<< ATROPELLO >>":GOSUB 1430:PRI
NT
1530 A$="EL CONDUCTOR DEBE ATROPELLAR AL
":GOSUB 1430:A$="TRANSEUNTE ANTES DE AGO
TARSE EL TIEMPO":GOSUB 1430
1540 PRINT:A$="CONTROLES:":GOSUB 1430:A$
="JUG. IZQUIERDO        JUG. DERECHO":GOSUB
 1430:A$="PEATON          CONDUCTOR":GOSUB
 1430:A$="Q.A.X.C.              CURSORES":GO
SUB 1430
1550 PRINT:a$="PULSA -S- PARA EMPEZAR":G
OSUB 1430
1560 IF INKEY(60)<>0 THEN 1560
1570 REM
1580 REM ******************************
*****
1590 REM * INICIALIZAR VARIABLES Y PANTA
LLA *
1600 REM ******************************
*****
1610 REM
1620 PAPER 0:INK 0,22:BORDER 22:MODE 1
1630 WINDOW #1,2,39,1,1:WINDOW 1,40,5,24
1640 INK 1,0:INK 2,6:INK 3,8
1650 PEN 2:GOSUB 2160
1660 T=250:GOSUB 2090
1670 P$="f":XP=19:YP=19:XL=3:YL=4
1680 PX=XP:PY=YP:LY=YL:LX=XL
1690 REM
1700 REM ******************
1710 REM * BUCLE PRINCIPAL *
1720 REM ******************
1730 REM
1740 IF NOT INKEY(0) THEN XP=XP-1:P$="e"
1750 IF NOT INKEY(2) THEN XP=XP+1:P$="e"
1760 IF NOT INKEY(1) THEN YP=YP+1:P$="f"
1770 IF NOT INKEY(8) THEN YP=YP-1:P$="f"
```

—La presentación incluye dos apartados; la portada de presentación, líneas 1000 a 1460, que ha sido creada en el MODO 0 de pantalla, y las instrucciones líneas 1470-1560.

De este bloque cabe destacar el modo en que se imprimen las instrucciones, que se hacen cargando en la variables A$, la frase a imprimir, y llamando con un GOSUB a la subrutina encargada de centrar el texto, y de imprimirlo letra a letra, situada en las líneas 1430-1460. La forma en que funciona esta subrutina es muy sencilla, con la función LEN calculamos la longitud de la cadena A$, y como sabemos que la pantalla en MODO 1 tiene de ancho 40 caracteres, le restamos la longitud de la cadena y el número resultante dividido entre 2 son los espacios que hay que dejar empleando TAB antes de imprimir la frase.

Para imprimir la línea de texto, se hace un bucle que tome desde el primer elemento de la cadena hasta el último (calculado con LEN (A$)), y subdividendo la cadena con MID$ vamos imprimiendo una a una las letras.

—El segundo bloque del que hemos hablado, el juego, es tal



```
1780 LOCATE YP,XP:CALL 30000:C=PEEK(2999
9)
1790 IF C=97 THEN XP=PX:YP=PY
1800 IF C=99 THEN GOTO 1990
1810 PEN 1:LOCATE PY,PX:PRINT" ";:LOCATE
 YP,XP:PRINT P$;
1820 PX=XP:PY=YP:SOUND 1,200,6,10,0,0,31
1830 IF NOT INKEY(67) THEN XL=XL-1
1840 IF NOT INKEY(69) THEN XL=XL+1
1850 IF NOT INKEY(62) THEN YL=YL+1
1860 IF NOT INKEY(63) THEN YL=YL-1
1870 LOCATE YL,XL:CALL 30000:C=PEEK(2999
9)
1880 IF C=97 THEN XL=LX:YL=LY
1890 IF C=101 OR C=102 THEN GOTO 1990
1900 PEN 3:LOCATE LY,LX:PRINT" ";:LOCATE
 YL,XL:PRINT"c";
1910 LX=XL:LY=YL
1920 T=T-1:GOSUB 2090
1930 GOTO 1740
1940 REM
1950 REM *********************
1960 REM * GANA EL CONDUCTOR *
1970 REM *********************
1980 REM
1990 FOR A=100 TO 200:SOUND 1,A,2:NEXT
2000 MODE 1
2010 A$="EL CONDUCTOR HA ATROPELLADO":GO
SUB 1430:A$="AL TRANSEUNTE":GOSUB 1430
2020 PRINT:A$="GANA EL CONDUCTOR":GOSUB
1430:GOTO 2650
2030 GOTO 2030
2040 REM
2050 REM *********************
2060 REM * IMPRIME EL TIEMPO *
2070 REM *********************
2080 REM
2090 LOCATE #1,14,1:PRINT#1,"TIEMPO: ";U
SING "###";T:IF T<1 THEN 2630
2100 RETURN
2110 REM
2120 REM **************************
2130 REM * IMPRESION DEL CIRCUITO *
2140 REM **************************
2150 REM
2160 PRINT"aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaa";
2170 PRINT"aa
          aa";
2180 PRINT"a
           a";
2190 PRINT"a      aaaaaaaaaaaaaaaaaaaaaaa
aaaaaaa    a";
```

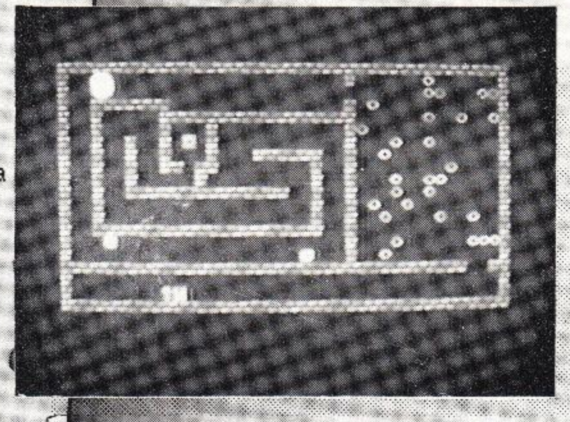
vez la parte más complicada, pero vamos con ello.

Lo primero que se hace es fijar el modo de pantalla, definir el color del borde, del papel y de las tintas, definir dos ventanas a inicializar las variables que se van a usar durante el juego. Esto comprende las líneas 1570-1680.

En el grupo de líneas 1690-1930 se encuentra el bucle principal. En este bucle lo primero que se hace es comprobar qué teclas son pulsadas por el jugador de la derecha. Inmediatamente despues, mediante el empleo de una rutina en máquina (más adelante la explicaremos), comprobaremos si el coche choca contra el muro, entonces vuelve a las coordenadas anteriores, o si choca contra el peatón, lo que significa que lo ha atropellado por lo que se salta la línea 1990 que da como vencedor al conductor. Luego se imprime el coche y se produce el sonido de su motor.

A continuación se leen las teclas pulsadas por el peatón, y al igual que antes comprobamos si choca contra los ladrillos o contra el coche, y lo imprimimos.

El siguiente paso es decrementar en una unidad el tiem-



```
2200 PRINT"a  aa
       aa  a";
2210 PRINT"a  a
        a  a";
2220 PRINT"a  a   aaaaaaaa  aaaaaaaaaaa
aaa    a  a";
2230 PRINT"a  a  aa
  aa   a  a";
2240 PRINT"a  a  a
  a   a  a";
2250 PRINT"a  a  a   aaaaa  aaaaaaaaaaa
  a   a  a";
2260 PRINT"a  a  a   aaaaa  aaa        aa
a  a   a  a";
2270 PRINT"a  a  a             a        a
  a   a  a";
2280 PRINT"a  a  aa            a   aaa   a
  aa   a  a";
2290 PRINT"a  a   aaaaaaaa  a  aaaaa  a
  a   a  a";
2300 PRINT"a  a               aa      a  a
        a  a";
2310 PRINT"a  aa               a       a  a
a    aa  a";
2320 PRINT"a   aaaaaaaa   aaaaaaaaaa
aaaaaaa   a";
2330 PRINT"a                a
          a";
2340 PRINT"aa              aa
        aa";
2350 PRINT"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaa";
2360 RETURN
2370 REM
2380 REM ***************************
2390 REM * C/M RECONOCER CARACTERES *
2400 REM ***************************
2410 REM
2420 DATA 205,96,187,50,47,117,201
2430 RESTORE 2420
2440 FOR a=30000 TO 30006:READ b:POKE a,
b:NEXT a
2450 REM
2460 REM *********************
2470 REM * REDEFINIR GRAFICOS *
2480 REM *********************
2490 REM
2500 SYMBOL AFTER 97
2510 SYMBOL 97,0,127,127,127,0,247,247,2
47
2520 SYMBOL 98,0,144,158,98,4,72,40,16
2530 SYMBOL 99,28,28,8,62,93,93,20,54
2540 SYMBOL 100,74,0,34,136,33,136,1,36
```

po, y llamar a la subrutina que lo imprime y comprueba si se ha acabado, localizada en las líneas 2040-2100. La siguiente línea, la 1930, hace un GOTO 1740 que vuelve a repetir todo el proceso.

— El tercer bloque, la conclusión, consta de tres partes:

Informe de que ha ganado el conductor líneas 1940-2020.

Informe de que el ganador es el peatón, líneas 2580-2640.

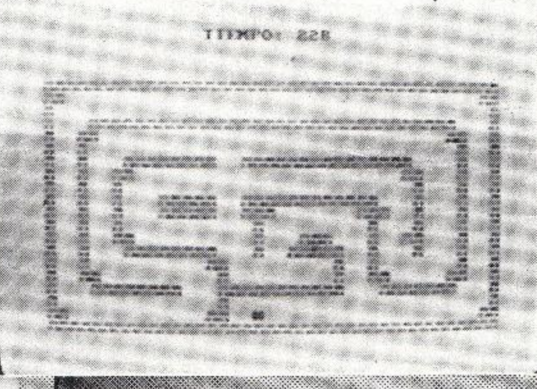
Y el final propiamente dicho, donde se pregunta si se desea jugar de nuevo. Líneas 2650-2790.

Hemos dejado para el final ciertas subrutinas, que son las siguientes:

La que imprime el circuito. Líneas 2110-2360.

Introducción en memoria de los datos de la rutina en C/M. Líneas 2370-2440. Su listado en ensamblador sería:

```
ORG 30000
CALL # BDD6; salto directo al firware. Lee carácter.
LD (29999),A; guarda en 29999 el carácter leido.
RET
```

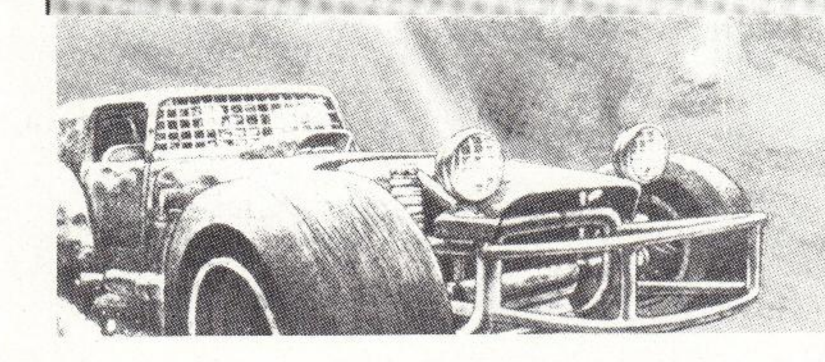


```
2550 SYMBOL 101,60,126,102,60,60,102,126
,60
2560 SYMBOL 102,0,102,255,219,219,255,10
2,0
2570 RETURN
2580 REM
2590 REM ******************
2600 REM * GANA EL PEATON *
2610 REM ******************
2620 REM
2630 MODE 1:LOCATE 1,5:A$="EL PEATON HA
LOGRADO ESQUIVAR":GOSUB 1430:A$="AL COND
UCTOR":GOSUB 1430:PRINT:A$="GANA EL PEAT
ON":GOSUB 1430
2640 GOTO 2650
2650 REM
2660 REM **********************
2670 REM * FINALIZA EL JUEGO *
2680 REM **********************
2690 REM
2700 FOR A=1 TO 20:FOR B=100 TO 200 STEP
 A:SOUND 1,B,1,15,0,0,31:NEXT B:NEXT A
2710 INK 0,0:INK 1,26:INK 2,15,0:PAPER 0
:BORDER 0:PEN 1:MODE 0
2720 PEN 2:LOCATE 6,7:PRINT"GAME OVER"
2730 PEN 1:LOCATE 4,12:PRINT"TIEMPO:";US
ING "######";T
2740 PEN 3:INK 3,15:LOCATE 5,17:PRINT "O
TRA? (S/N)"
2750 FOR A=1 TO 50:A$=INKEY$:NEXT A
2760 PEN 1
2770 IF INKEY(60)=0 THEN RUN 1410
2780 IF INKEY(46)=0 THEN CALL 0
2790 GOTO 2770
```

Esta sencilla rutina se basa en una llamada al sistema operativo, a la subrutina TXT UNWRITE, que lee un carácter de la pantalla. H debe contener la columna, L contiene la fila, y tras la llamada en A se devuelve el código del carácter, que se guarda en la posición de memoria 29999 para ser leido desde el BASIC.

—Redefinición de los gráficos. Líneas 2450-2570. El método empleado, es el normal; SIMBOL AFTER n para decir a partir de que gráfico se va ha cambiar, y SYMBOL n, 8 datos, para ir definiendo los gráficos.

Si eres programador pueden serte de utilidad la subrutina que centra, y la que reconoce el carácter que hay en la posición del cursor.



MOSQUITOS

```
10 REM ****** MOSQUITOS ******
20 REM * ANGEL GARCIA DELGADO *
30 REM ************************
40 MODE 0
50 INK 0,0:INK 1,26:INK 2,7:INK 3,18
60 PAPER 0:BORDER 0
70 ORIGIN 318,210
80 FOR a=0 TO 2*PI STEP PI/40
90 PLOT 0,0,2
100 DRAW 80*COS(a),80*SIN(a)
110 DRAWR 300*COS(a),300*SIN(a),3
120 NEXT a
130 PRINT CHR$(22)+CHR$(1);
140 LOCATE 7,12
150 PRINT "MOSQUITOS"
160 PRINT CHR$(22)+CHR$(0);
170 INK 2,7,0:INK 3,0,18
180 FOR a=100 TO 0 STEP -1:SOUND 1,a,1,1
5:NEXT a
190 INK 2,18,7:INK 3,7,18
200 FOR a=300 TO 1 STEP -2:SOUND 1,RND*2
00,1:SOUND 2,RND*200,1:NEXT a
210 INK 2,7:INK 3,18
220 FOR a=1 TO 500:NEXT a
230 FOR a=0 TO 30:PRINT:FOR b=1 TO 50:NE
XT b:NEXT a
240 MODE 1
250 GOTO 300
260 PRINT TAB(20-LEN(a$)/2):FOR a=1 TO L
EN(a$)
270 b$=MID$(a$,a,1):PRINT b$;:IF b$<>" "
 THEN SOUND 1,a,2,15
280 NEXT a
290 PRINT:PRINT:FOR a=1 TO 200:NEXT a:RE
TURN
300 REM *** INICIO DEL PROGRAMA ***
310 A$="<<< MOSQUITOS >>>":GOSUB 260
320 PRINT:A$="CON LA LLEGADA DEL VERANO
VUELVEN":GOSUB 260
330 A$="LOS INDESEABLES MOSQUITOS. ELIMI
NA":GOSUB 260
340 A$="A LOS MOSQUITOS PEQUENNOS, PERO
SIN":GOSUB 260
350 A$="TOCAR A LOS MAS GRANDES":GOSUB 2
60
360 PRINT:A$="Q. ARRIBA":GOSUB 260
370 A$="A. ABAJO":GOSUB 260
380 A$="O. IZQUIERDA":GOSUB 260
390 A$="P. DERECHA":GOSUB 260
400 PRINT:A$="PULSA -S- PARA EMPEZAR":GO
SUB 260
```

Con la llegada del verano vuelven los indeseables mosquitos. Deberás eliminar a los mosquitos pequeños sin tocar a los grandes.

Controles:

Cursor arriba - arriba
Cursor abajo - abajo
Cursor izq. - izquierda
Cursor dcha. - derecha

La presentación del juego ha sido hecha en el MODO 0 de pantalla, y abarca las líneas 10-230. Se fijan los colores de las tintas, se posiciona el origen de coordenadas en el centro de la pantalla. Luego en la línea 130 se activa el modo transparente para imprimir el título del juego sobre el gráfico de la pantalla, y en la línea 160 se desactiva. A continuación para darle más vistosidad, se ponen parpadeantes las tintas, y se producen ciertos sonidos al azar.

En el grupo de líneas 240-410, se imprimen las instrucciones del juego, y las teclas necesarias para el movimiento. En la línea 410 se espera la pulsación de la tecla -S- para continuar.

```
410 IF INKEY(60)<>0 THEN GOTO 410
420 INK 0,22:INK 1,0:INK 2,8:INK 3,?
430 PAPER 0:BORDER 22:PEN 1
440 MODE 1
450 GOSUB 1150
460 GOSUB 1090
470 GOSUB 780
480 GOSUB 1010
490 P=0:GOSUB 1070
500 H=10:I=10
510 HB=10:IB=10
520 IF INKEY(67)=0 THEN H=H-1
530 IF INKEY(69)=0 THEN H=H+1
540 IF INKEY(34)=0 THEN I=I-1
550 IF INKEY(27)=0 THEN I=I+1
560 LOCATE I,H:CALL 30000:C=PEEK(29999)
570 PEN 1
580 LOCATE IB,HB:PRINT " "
590 LOCATE I,H:PRINT CHR$(161)
600 SOUND 1,2000,1,8
610 IB=I:HB=H
620 IF C>161 THEN GOTO 650
630 IF C=160 THEN FOR A=1 TO 200 STEP 20
:SOUND 1,A,1:NEXT A:P=P+15:GOSUB 1070:GO
SUB 930
640 GOTO 520
650 REM *** MUERTE ***
660 INK 0,6,12:BORDER 6,12
670 FOR A=1 TO 1000 STEP 3
680 SOUND 1,A,1
690 NEXT A
700 INK 0,22:BORDER 22
710 MODE 0
720 PEN 2:LOCATE 6,7:PRINT"GAME OVER"
730 PEN 1
740 LOCATE 4,13:PRINT"PUNTOS:";USING "##
####";P
750 INK 2,22,1
760 FOR A=1 TO 10000:NEXT A
770 RUN 240
780 MODE 1
790 WINDOW #1,1,40,1,3
800 PAPER #1,3:CLS #1
810 WINDOW 1,40,4,25
820 PEN 3
830 A$=CHR$(166)+CHR$(166)
840 B$=""
850 FOR A=1 TO 20
860 B$=B$+A$
870 NEXT A
880 PRINT B$;B$;
```

Comienza ahora el juego propiamente dicho. Lo primero que hacemos es poner la pantalla en MODO 1 y dar nuevos colores a las tintas.

En la línea 450 se hace una llamada a una subrutina (líneas 1150-1230) que se encarga de refefinir los nuevos gráficos.

En la siguiente línea de programa, la 460, se llama a otra subrutina localizada en el grupo de líneas 1090-1140, que introduce en memoria, a partir de la dirección 30.000, una rutina en C/M que nos devolverá en la posición de memoria 29999, el código del carácter que se halle en las coordenadas del cursor. Esta rutina sólo ocupa 7 bytes, puesto que lo único que hace es una llamada a una subrutina del sistema operativo.

La línea 470 es otra llamada a la subrutina (líneas 780-910) que dibuja el escenario. Se definen dos ventanas, una donde se imprimen los marcadores, y otra donde se desarrolla el juego.

En la línea 480 se dibuja el primer mosquito pequeño que habrá en pantalla, empleando la subrutina de las líneas 1010-1060.

A continuación se inicializa el marcador de puntos a cero, y se imprime el marcador llamando a la subrutina de las líneas 1070-1080.

El bucle principal abarca las líneas 500-640. Comienza inicializando las coordenadas de nuestro punto de mira.

```
890 FOR A=1 TO 17:PRINT A$;TAB(39);A$;:N
EXT A
900 PRINT B$;B$;
910 RETURN
920 REM *** MOSQUITO GRANDE ***
930 X=3+INT(RND*16):Y=2+INT(RND*35)
940 PEN 2
950 LOCATE Y,X:CALL 30000:C=PEEK(29999):
IF C<>32 THEN GOTO 930
960 LOCATE Y+1,X:CALL 30000:C=PEEK(29999
):IF C<>32 THEN GOTO 930
970 LOCATE Y+1,X+1:CALL 30000:C=PEEK(299
99):IF C<>32 THEN GOTO 930
980 LOCATE Y,X+1:CALL 30000:C=PEEK(29999
):IF C<>32 THEN GOTO 930
990 LOCATE Y,X:PRINT CHR$(162)+CHR$(163)
1000 LOCATE Y,X+1:PRINT CHR$(164)+CHR$(1
65)
1010 X=3+INT(RND*17):Y=2+INT(RND*36)
1020 LOCATE Y,X:CALL 30000:C=PEEK(29999)
1030 IF C<>32 THEN GOTO 1010
1040 SOUND 1,300,1
1050 PEN 2:PRINT CHR$(160)
1060 RETURN
1070 LOCATE #1,10,2:PRINT#1,"PUNTUACION:
";USING "######";P
1080 RETURN
1090 REM ** CODE SCREEN$ **
1100 DATA 205,96,187,50,47,117,201
1110 RESTORE 1100
1120 FOR a=30000 TO 30006:READ b:POKE a,
b:NEXT a
1130 RETURN
1140 REM ******************
1150 SYMBOL AFTER 160
1160 SYMBOL 160,80,168,156,223,120,36,68
,130
1170 SYMBOL 161,24,24,8,227,199,16,24,24
1180 SYMBOL 162,52,74,74,133,130,131,227
,63
1190 SYMBOL 163,0,0,0,48,248,255,248,192
1200 SYMBOL 164,15,20,20,44,216,16,96,0
1210 SYMBOL 165,128,192,160,176,64,48,0,
0
1220 SYMBOL 166,182,109,219,182,109,219,
182,109
1230 RETURN
```

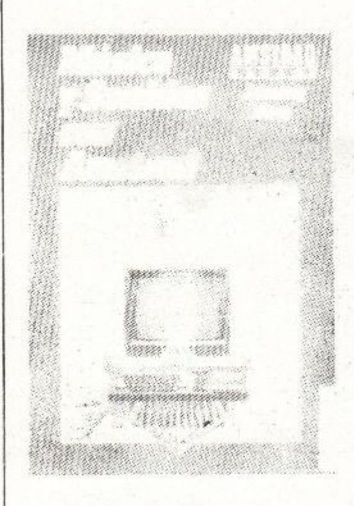
Luego se lee el teclado, líneas 520-550. Antes de imprimir nuestra mira, y empleando la rutina en máquina de la que hemos hablado, guardamos en la variable C, el carácter que había en la pantalla para luego tras imprimirla, comprobar si hemos chocado contra los muros, contra un mosquito pequeño, o contra un carácter de un mosquito de los grandes.

Si no chocamos, el programa continúa repitiendo el bucle, pero en caso contrario bifurca a la línea 650 donde se imprime el GAME OVER y la puntuación. Tras una larga pausa, se ejecuta de nuevo el programa a partir de la línea 240.



MUSICA Y SONIDOS CON AMSTRAND

Autor: Jeremy Vine. Editorial: Indescomp. Páginas: 80



Este libro va dirigido a que aprendamos a sacarle el máximo partido a las magníficas posibilidades sonoras del Amstrad.

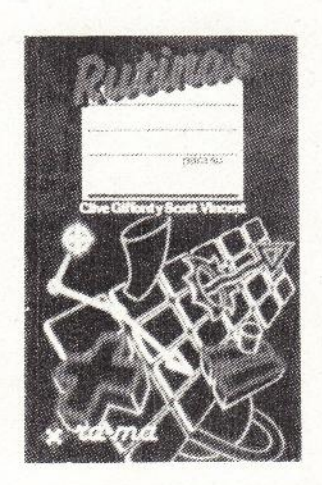
El libro tiene dos objetivos principales. En primer lugar, enseñar los rudimentos de la generación del sonido y la música. Sobre este tema hay una amplia información que permitirá obtener los conocimientos necesarios para programar música, aún cuando el usuario no sepa tocar ningún instrumento musical ni leer partituras.

El segundo objetivo del libro es explicar las instrucciones que controlan los efectos sonoros; Para lo cual se da un repaso a las interioridades del generador de sonido del Amstrad y se muestran una serie de programas y tablas que nos permitirán aprender de una forma eminentemente práctica.

Con todo, se ofrecen programas ejemplo de sonidos de todo tipo, tales como, timbres de teléfono, explosiones, alarmas, sirenas, escalas musicales, campanas y silbidos, con el objeto de inducir a experimentar y descubrir efectos especiales propios.

RUTINAS EN CODIGO MAQUINA PARA SU AMSTRAD

Autor: Clive Gifford y Scott Vincent. Editorial: Rama. Páginas: 92



Este libro es para aquellos usuarios que escriben programas en Basic pero todavía no se han introducido en el complicado Código Máquina, o aún desconocen muchas de sus rutinas.

Se describen 20 rutinas que tratan sobre gráficos, sonido, conservación de memoria, recuperación más rápida sobre el cassette y utilidades. Todas van suministradas en la forma de un cargador Basic, un listado en Ensamblador y una descripción detallada del uso; además en algunos casos se incluye un programa de demostración.

No dudamos que con este libro podréis conseguir extraordinarios efectos tales como suaves desplazamientos de la pantalla, rotación de caracteres multicolores, nuevos sonidos, caracteres itálicos y otros efectos-rutinas que en Basic son imposibles de realizar, y que mejorarán notablemente vuestros programas, tanto si son juegos educativos o aplicaciones.



¡Tu cinta para grabar guay!
HOLIDAY
HOLIDAY C-40
HOLIDAY C-60
Tu cinta virgen de 40, 60 y 90
HOLIDAY C-90
V. CARRERA
Fabricada por IBEROFON, S.A.
Avenida de Fuentemar, 35 - Polígono Industrial de Coslada - MADRID
Teléfonos 671 22 00-04-08-12 - Télex 42797 FONO E - Telefax (91) 671 39 09
S ON S
Direct Metal DMM Mastering
COMPACT disc DIGITAL AUDIO
discoflex®


SPECTRUM • AMSTRAD CPC • COMMODORE • MSX
FREDDY HARDEST
FICHA TECNICA
• Dominio de las artes marciales: Patada - Puñetazo - Protección.
• Turbo-Laser de carga atmosférica.
• Salto controlable en altura y dirección.
• 30 Pantallas de Multiscroll.
• 64 Pantallas conectadas por ascensores y túneles.
• 16 Terminales de ordenador para hacer de Hacker.
• Enemigos inteligentes que se protegen de tu Laser.
• 5 Especies de animales hostiles.
• 3 Tipos de Androides asesinos.
FX DOBLE CARGA
Signfica dos programas en uno, el doble de acción, el doble de aventura, el doble de gráficos y máxima adictividad. Prueba un FX DOBLE CARGA de DINAMIC, te sorprenderá.
SPECTRUM
COMMODORE
AMSTRAD CPC
MSX
FREDDY HARDEST, agente secreto y refutado playboy, se encuentra en un satélite enemigo con su nave averiada. Para escapar, deberá llegar hasta la base enemiga y robar un caza. FREDDY, demostrará su destreza saltando, agarrándose a las argollas que encuentre, trepando por cuerdas, disparando su Laser... FREDDY, es todo un número uno, pero sobre todo, quiere escapar vivo.
DINAMIC
DINAMIC SOFTWARE. PZA. DE ESPAÑA, 18. TORRE DE MADRID, 29-1. 28008 MADRID. TELEX: 44124 DSOFT-E
TIENDAS Y DISTRIBUIDORES: (91) 314 • 18 • 04. PEDIDOS CONTRA REEMBOLSO: (91) 248 • 78 • 87.
875